

信号与信息处理
——技术丛书

数字版权 保护技术及其应用

冯耕平 编著

信号与信息处理技术丛书

数字版权保护技术及其应用

冯柳平 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书重点讲述数字版权保护技术及其应用。全书共分9章,包括数字版权管理技术、加密技术与数字签名技术、数字水印技术、数字指纹技术、DRM 标准、权利描述语言、DRM 应用、印刷品防伪技术和抗几何攻击的数字水印算法。

本书内容丰富、层次清晰,可作为高等院校本科、研究生各相关专业的教材,也适合学习数字版权保护技术的人员参考使用。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

数字版权保护技术及其应用 / 冯柳平编著. —北京: 电子工业出版社, 2013.8

(信号与信息处理技术丛书)

ISBN 978-7-121-19803-8

I. ①数… II. ①冯… III. ①电子出版物—版权—保护—研究 IV. ①D913.04

中国版本图书馆 CIP 数据核字(2013)第 046810 号

责任编辑: 董亚峰 特约编辑: 王 纲

印 刷: 涿州市京南印刷厂

装 订: 涿州市京南印刷厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 22 字数: 565 千字

印 次: 2013 年 8 月第 1 次印刷

定 价: 48.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前言

随着互联网和数字化技术的快速发展，网上交易和传播的电子书、音乐、电影、图片、游戏和软件等数字内容越来越多，数字出版物的读者群已经初步形成，数字出版成为出版业未来的发展趋势。但是，由于信息的复制更加快捷简便，盗版现象日益严重，给相关权利人造成巨大的经济损失，挫伤他们使用互联网扩展业务的积极性，并直接威胁数字出版业的健康、可持续发展。

面对有巨大潜力的数字出版市场，如何保护数字作品的版权已成为近年来法律界和 IT 业界的热点问题，同时也是难点问题。传统的版权保护模式已不能满足数字内容版权保护的需要，人们对数字版权保护提出了新的要求，数字版权管理（Digital Rights Management，DRM）技术在这种背景下应运而生。目前常用的数字版权保护技术还有加密技术和数字签名技术、数字水印技术和数字指纹技术等。

DRM 技术是数字网络环境下数字内容交易和传播的重要技术。本书介绍了数字出版领域版权保护的关键技术、相关的标准和应用以及最新进展，使读者在掌握数字版权保护技术基本内容的基础上，对该领域未来的发展趋势及关键技术有所了解。

本书共分为 9 章，第 1 章数字版权管理技术，主要介绍 DRM 系统模型及关键技术，第 2~4 章介绍 DRM 系统的底层技术，包括加密技术与数字签名技术、数字水印技术和数字指纹技术。其中，第 2 章除了介绍常规的加密技术与数字签名技术，还对 DRM 加密技术进行了概述；第 3 章介绍了数字水印的基本概念、分类及性能指标，并对空域和频域主要的数字水印算法和鲁棒性测试软件——StirMark 基准测试程序进行了描述；第 4 章介绍数字指纹的系统模型、数字指纹编码和数字指纹协议。第 5 章主要介绍 OMA DRM 标准，并对 AVS DRM 标准进行了概述。第 6 章主要介绍当前发展最为完善的两个基于 XML 的权利描述语言——XrML 语言和 ODRL 语言，并对基于逻辑的权利描述语言——LicenseScript 语言的基本内容进行了介绍。目前 DRM 技术的应用领域主要是电子书、流媒体、电子文档等，第 7 章介绍了在这几个领域主要的 DRM 应用。第 8 章印刷品防伪技术，这是数字水印在印刷领域的应用，分析了在打印扫描过程中图像所受到的攻击，包括像素点失真和几何变形，并介绍抗打印扫描的数字水印算法。第 9 章介绍抗几何攻击的数字水印算法，分析了几何攻击对数字水印系统的影响，重点介绍了第二代数字水印——基于图像特征的水印算法，主要包括基于 Harris 特征点和 SIFT 特征点的方法。

本书可作为高等学校信号与信息处理相关专业高年级学生及硕士研究生信息安全技术课程的教材，也可用做信号与信息处理相关专业或相关领域研究人员的参考书。本书约需

40~60 学时讲授。

在本书的编写过程中，参考了国内外有关的数字版权保护技术的众多文献，特别是国内外的研究生学位论文，对某些内容进行了较好的综述与算法描述，参考过的论文在每章的参考文献中已列出，在此对所有参阅与引用了的文献与论文作者表示衷心感谢。

本书得到了高端印刷装备信号与信息处理北京市重点实验室师生的大力支持，尤其是曹鹏教授在本书编写过程中提出了有益的建议，在此表示衷心感谢。研究生徐佳和闻爱华在书稿编排、画图、校对过程中做了大量工作，在此一并致谢。

本书得到了国家自然科学基金项目（No. 61170259）和北京印刷学院重点研究项目（No.E-a-2013-20）的资助，在此特表感谢。

由于编者水平有限，加上数字版权保护技术本身在不断丰富和发展，尽管数易其稿，但书中难免存在不妥乃至错误之处，敬请读者不吝指正。

编 者
2013 年 6 月

目 录

第 1 章 数字版权管理技术	1
1.1 DRM 系统模型	2
1.2 数字唯一对象标识	4
1.2.1 DOI 在 DRM 系统中的作用	4
1.2.2 DOI 系统概述	5
1.2.3 DOI 的语法	5
1.2.4 DOI 的解析	6
1.3 DRM 系统中数字内容的使用控制	7
1.3.1 用户控制	7
1.3.2 权利描述与控制	9
1.4 权利转移	11
1.4.1 权利迁移与共享	11
1.4.2 二次分发	13
1.5 可信执行	15
1.5.1 DRM 的安全性	15
1.5.2 可信平台模块	16
1.5.3 可信平台的信任链度量机制	17
1.6 互操作性	19
1.6.1 DRM 系统互操作的现状	19
1.6.2 Coral DRM 互操作框架	20
1.6.3 面向服务的框架	21
1.6.4 DRM 内容改写体制	22
参考文献	23
第 2 章 加密技术与数字签名技术	26
2.1 密码学概述	27
2.1.1 密码体制与密码系统的基本模型	27
2.1.2 Kerckhoff 假设和密码系统的安全性	27
2.1.3 分组密码的分析方法	28
2.2 对称密码体制	29

2.2.1	分组密码的设计思想与 Feistel 密码结构	29
2.2.2	数据加密标准	31
2.2.3	高级加密标准	37
2.3	公钥密码体制	43
2.3.1	公钥密码的基本思想	43
2.3.2	背包加密算法	44
2.3.3	RSA 算法	46
2.3.4	ElGamal 算法	48
2.3.5	椭圆曲线加密算法	50
2.4	消息认证	53
2.4.1	消息认证码	53
2.4.2	Hash 函数	53
2.4.3	MD5 算法	55
2.4.4	SHA 算法	58
2.5	数字签名	60
2.5.1	数字签名概述	60
2.5.2	数字签名标准	62
2.6	特殊的数字签名	63
2.6.1	盲签名	63
2.6.2	群签名	64
2.7	PKI 认证体系	67
2.7.1	PKI 的概念	67
2.7.2	PKI 的组成	67
2.7.3	PKI 的标准	68
2.7.4	认证中心	70
2.7.5	数字证书	71
2.8	DRM 加密	73
2.8.1	DRM 加密概述	73
2.8.2	DRM 加密的结构	73
2.8.3	DRM 加密算法	76
2.8.4	DRM 加密效果的检验	81
	参考文献	82
第 3 章	数字水印技术	85
3.1	数字水印概述	86
3.1.1	数字水印的系统模型	86
3.1.2	数字水印的分类	87
3.1.3	数字水印的性能分析	88
3.2	空域图像水印算法	89

3.3 DCT 域水印算法 91

3.3.1 离散余弦变换的基本概念 91

3.3.2 基于 DCT 变换的水印嵌入和提取算法 92

3.4 DWT 域水印算法 93

3.4.1 小波变换的基本概念 94

3.4.2 数字图像的离散小波变换 95

3.4.3 基于 DWT 的水印算法 96

3.5 Contourlet 域水印算法 98

3.5.1 Contourlet 变换 98

3.5.2 基于 Contourlet 变换的水印算法 100

3.6 水印攻击 101

3.6.1 鲁棒性攻击 101

3.6.2 表达攻击 103

3.6.3 解释攻击 104

3.7 Stirmark 基准测试程序 105

3.7.1 Stirmark 概述 105

3.7.2 用户 API 接口 106

3.7.3 配置测试方案 106

3.7.4 执行测试程序 108

参考文献 108

第 4 章 数字指纹技术 111

4.1 数字指纹的基本概念 112

4.1.1 数字指纹的系统模型 112

4.1.2 数字指纹方案的基本要求 113

4.2 数字指纹编码 114

4.2.1 合谋攻击 114

4.2.2 连续指纹编码 115

4.2.3 c-安全码 116

4.2.4 BIBD 编码 117

4.2.5 基于残留特征跟踪的指纹编码 120

4.3 数字指纹协议 124

4.3.1 对称数字指纹协议 124

4.3.2 非对称指纹协议 126

4.3.3 匿名指纹 131

参考文献 135

第 5 章 DRM 标准 138

5.1 OMA DRM 1.0 139

5.2	OMA DRM 2.0 体系结构	141
5.2.1	角色定义	141
5.2.2	OMA DRM 2.0 的基本架构	142
5.2.3	OMA DRM 2.0 工作机制	143
5.3	ROAP	145
5.3.1	ROAP 的工作流程	145
5.3.2	域与非连接设备支持	147
5.3.3	超级分发	149
5.3.4	流媒体的支持	150
5.4	OMA DRM 2.0 内容格式	150
5.4.1	基础数据结构定义	150
5.4.2	DCF	153
5.4.3	PDCF	154
5.5	OMA DRM 2.0 权利描述	156
5.6	OMA DRM 2.0 安全机制	162
5.7	AVS DRM 标准	164
5.7.1	AVS 标准概述	164
5.7.2	AVS DRM 核心档	166
5.7.3	AVS DRM 权利描述	166
5.7.4	AVS DRM 网络电视档	167
	参考文献	169
第 6 章	权利描述语言	170
6.1	XrML 的数据模型	171
6.1.1	数据模型中的实体	171
6.1.2	实体之间的关系	172
6.2	数据模型在 XML Schema 中的封装	174
6.2.1	XrML 的组织结构	174
6.2.2	强制项和可选项	175
6.2.3	核心模式	176
6.2.4	标准扩展模式	177
6.2.5	内容扩展模式	179
6.3	核心模式的基本语法	181
6.3.1	主体	181
6.3.2	权限	183
6.3.3	资源	184
6.3.4	条件	186
6.3.5	其他内核类型和元素	189
6.4	XrML 的运行机制	192

6.4.1	XrML SDK 结构	192
6.4.2	基本流程	193
6.4.3	条件验证器行为状态转换机制	193
6.4.4	条件验证工作流程	194
6.5	XML 加密	195
6.5.1	XML 安全标准概述	195
6.5.2	XML 加密和传统加密的区别	196
6.5.3	XML 加密规范和基本结构	197
6.5.4	XML 加密粒度的选择	199
6.6	XML 数字签名	203
6.6.1	XML 签名概述	203
6.6.2	XML 签名的基本结构和语法	203
6.6.3	创建 XML 签名	205
6.6.4	验证 XML 签名	206
6.7	ODRL	206
6.7.1	ODRL 模型	206
6.7.2	ODRL 安全模型	218
6.7.3	ODRL 表达式	222
6.7.4	ODRL XML 语法	226
6.7.5	ODRL XML 例子	227
6.8	LicenseScript 简介	236
6.8.1	基于 XML 的权限描述语言存在的问题	236
6.8.2	许可证	237
6.8.3	重写规则	238
6.8.4	LicenseScript 执行模型	239
	参考文献	241
第 7 章	DRM 应用	243
7.1	流媒体的 DRM	244
7.1.1	流媒体介绍	244
7.1.2	WMRM	244
7.1.3	Helix DRM 方案	251
7.2	电子书的 DRM	254
7.2.1	电子书的发展概况	254
7.2.2	Microsoft 电子书系统	256
7.2.3	Adobe 电子书系统	257
7.2.4	方正 Apabi 电子书系统	258
7.2.5	电子书 DRM 应用方案的比较分析	260
7.3	电子文档的 DRM	261

7.3.1 电子文档的格式	261
7.3.2 基于 RMS 的 Microsoft Office 2003	262
7.3.3 Adobe 公司的 Adobe Acrobat	263
7.3.4 北大方正 Apabi 文档保护系统	264
7.4 开放源代码 OpenIPMP	266
参考文献	269
第 8 章 印刷品防伪技术	270
8.1 抵抗硬复制输出的数字水印技术	271
8.2 打印扫描过程中图像的畸变分析	273
8.2.1 像素点的失真分析	273
8.2.2 几何失真	274
8.3 基于频域系数的抗打印扫描水印算法	274
8.3.1 打印扫描在 DCT 域上对图像的影响	275
8.3.2 水印嵌入算法	277
8.3.3 水印提取算法	278
8.3.4 实验结果及分析	279
8.4 数字半色调技术	279
8.4.1 半色调技术概述	279
8.4.2 阈值抖动法	280
8.4.3 误差分散法	283
8.4.4 点分散法	285
8.4.5 噪声半色调法	285
8.4.6 影响数字半色调的因素	288
8.5 半色调数字水印技术	293
8.5.1 半色调水印技术的基本方法	293
8.5.2 核转换误差分散水印算法	296
8.5.3 半色调水印存在问题和研究前景	299
参考文献	300
第 9 章 抗几何攻击的数字水印算法	302
9.1 几何攻击	303
9.1.1 全局几何攻击	303
9.1.2 局部几何攻击	305
9.2 几何攻击对数字水印系统的影响	306
9.3 抗几何攻击的数字水印技术	308
9.3.1 基于几何校正的方法	308
9.3.2 基于几何不变域的方法	311

9.3.3 基于图像特征的方法 314

9.4 基于 Harris 特征点的抗几何攻击的数字图像水印算法 315

9.4.1 Harris 特征点检测 315

9.4.2 Delaunay 三角剖分 317

9.4.3 基于 Harris 特征点和 Delaunay 三角剖分的水印算法 317

9.5 基于 SIFT 特征点的抗几何攻击的数字图像水印算法 318

9.5.1 尺度空间理论 318

9.5.2 SIFT 特征点 319

9.5.3 基于 SIFT 的水印同步 322

9.5.4 基于 SIFT 特征点的 NSCT 域水印嵌入算法 325

9.5.5 基于 SIFT 特征点的 NSCT 域水印提取算法 329

9.5.6 实验结果与讨论 331

参考文献 335

数字版权管理技术

数字版权保护是近年来法律界和 IT 业界一个亟待解决的问题，第一代数字版权保护技术主要致力于对数字内容的安全性与加密技术的开发，在提供数字化和网络化信息服务的同时，有效地阻止对这些信息的非法使用和复制，以达到保护数字知识产权的目的。但随着数字出版的发展和广泛使用，采用传统的加密技术已不能满足数字版权保护的需要。为了更好地保护数字内容的版权，人们提出了一种新的技术——数字版权管理（Digital Rights Management, DRM）技术，即第二代数字版权保护技术，确保数字内容的合法使用和传播。

DRM 技术是在网络及数字化环境下，借助加密与封装技术、PKI 认证、权限管理技术等，使数字内容和权利主体获得对其客体的控制权，从而防止非授权使用，保护权利所有人利益的一种综合性技术体制。DRM 技术对数字内容版权的保护，贯穿数字内容从产生到分发、从销售到使用的整个内容流通过程。文献[1]对 DRM 技术进行了全面的综述。

在网络出版领域，DRM 技术的地位越来越重要。2001 年，DRM 技术被 MIT 的《Technology Review》杂志评为“将影响世界”十大新兴技术之一；随后在旧金山举行的 Seybold 上，DRM 成了一个技术热点，并且人们普遍认为，20 世纪 90 年代中期开始，属于 DRM 的实验阶段，发展到今天，DRM 已经成为一项很重要的技术，特别是数字媒体领域，DRM 成为了必需的技术。

美国计算机协会从 2001 年开始，每年举办一次“ACM Workshop on Digital Rights Management”会议，涉及的内容包括多个方面，主要有 DRM 系统的体系结构、DRM 中对数字内容使用的跟踪和审核、数字内容交易的商业模式及其安全性需求、多媒体数据的加密、身份识别、DRM 系统中的密钥管理、数字内容使用权利的转移问题、数字版权描述等。

越来越多的数字内容通过 DRM 技术保护，实现了数字内容的增值服务。例如，电子书通过 DRM 系统进行网上销售，或者把电子书卖给有 DRM 保护的数字图书馆，使出版社在出版纸书的同时，通过电子书的销售获得更多的收益。在电子书、电子报纸、电子杂志等数字内容的销售方面，DRM 技术发挥了重要的作用。随着移动数据增值业务的迅猛发展，内容提供商通过大量下载类业务及 MMS 等信息类业务传播的音视频和应用软件、游戏等数字内容越来越多，将 DRM 技术引入移动增值业务，可以确保数字内容在移动网内传播，保证内容提供商的利益。移动 DRM 已成为目前全球范围内移动业务研究的热点之一。

1.1 DRM 系统模型

不同的 DRM 系统虽然在所侧重的保护对象、支持的商业模式和采用的技术方面不尽相同，但是它们的核心思想是相同的，都是通过使用数字许可证来保护数字内容的版权。用户得到数字内容后，必须获得相应的数字许可证才可以使用该内容。

图 1-1 给出了典型的 DRM 系统参考体系结构，包括三个主要模块：内容服务器（Content Server）、许可证服务器（License Server）和客户端（Client）^{[1][2]}。

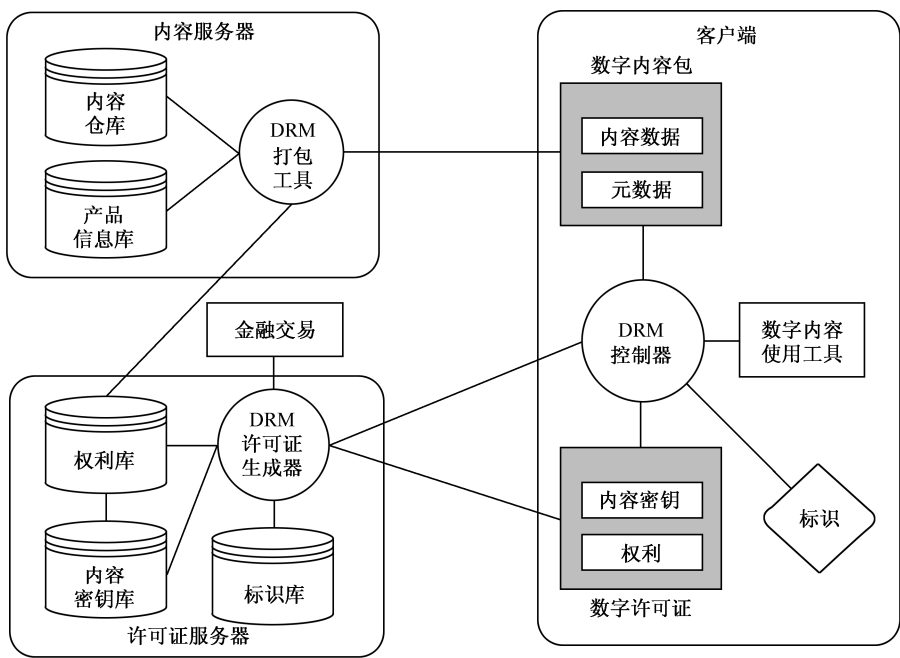


图 1-1 DRM系统参考体系结构

内容服务器通常包括存储数字内容的内容仓库、存储产品信息的产品信息库和对数字内容进行安全处理的 DRM 打包工具。该模块主要实现对数字内容的加密、插入数字水印等处理，并将处理结果和内容标识元数据等信息一起打包成可以分发售的数字内容。另外一个重要功能就是创建数字内容的使用权利，把数字内容密钥和使用权利信息发送给许可证服务器。

许可证服务器包含权利库、内容密钥库、用户身份标识库和 DRM 许可证生成器，经常由一个可信的第三方——清算中心负责。该模块主要用来生成并分发数字许可证，还可以实现用户身份认证、触发支付等金融交易事务。数字许可证是一个包含数字内容使用权利（包括使用权限、使用次数、使用期限和使用条件等）、许可证颁发者及其拥有者信息的计算机文件，用来描述数字内容授权信息，由权利描述语言描述。大多数 DRM 系统中，数字内容本身经过加密处理。因此，数字许可证通常还包含数字内容解密密钥等信息。

客户端主要包含 DRM 控制器和数字内容使用工具。DRM 控制器负责收集用户身份标识

等信息,控制数字内容的使用。如果没有许可证,DRM 控制器还负责向许可证服务器申请许可证。数字内容使用工具主要用来辅助用户使用数字内容。

当前大部分 DRM 系统都是基于该参考体系结构的,如 Microsoft WMRM、Inter Trust Rights System、Adobe Content Server、RealNetworks RMCS 和 IBM EMMS 等。通常情况下,DRM 系统还包括分发服务器和零售门户网站,特别是支持数字内容网上交易的 DRM 系统。分发服务器存放打包后的数字内容,负责数字内容的分发。零售门户网站直接面向用户,通常作为用户和分发服务器、版权服务器以及(金融)清算中心的桥梁,用户本身只与门户网站交互。

DRM 技术不是密码技术的简单应用,也不是将受保护的内容从服务器传递到客户端并用某种方式限制其使用的简单机制。内容提供者希望通过使用 DRM,保护数字作品的版权,促进数字化市场的发展。因此,用户对 DRM 系统的接受度也是必须考虑的。一个完善的 DRM 系统必须兼顾提供者和使用者双方的需求,具备以下功能^{[1][2]}。

① 提供管理、保护和跟踪数字内容的功能,只有合法的用户才可以使用数字内容。支持对各种形式使用权利的描述、识别、交易、保护、监控和跟踪。

② 提供透明易用的体验环境,保护用户的合法权益和隐私。使用者可以自由选择、购买数字内容,可以在多种设备上使用数字内容。在合法的范围内,可以不受时间、地点、网络状况的限制使用数字内容,可以转卖、赠送或者出借购买的数字内容,支持用户变更数字内容使用设备,支持法律规定的用于保护公共利益的相关权利,如合理使用(Fair Use)。

因此 DRM 需要解决的关键问题包括以下内容^{[1][2]}。

① 数字内容的安全性:保证数字内容在出版发行、分发、使用等整个流通过程中的安全性。数字内容的安全性是数字内容版权保护最基本的要求,主要包括数字内容的机密性、完整性和非否认性。

② 权利描述:描述数字内容授权信息,支持不同商业模式下各种数字内容各类使用权利的描述。

③ 使用控制:控制数字内容的使用,确保只有授权使用者才可以使用受保护的数字内容。同时,用户对数字内容只拥有授予的使用权利,根据使用权利对数字内容进行访问。

④ 合理使用:支持用户对数字内容的合理使用,平衡版权持有人和公众之间的利益。

⑤ 权利转移:支持数字内容使用权利的转移,可以转移到另外一台设备上,也可以暂时或永久地转移给其他用户,使得用户可以更换数字内容使用设备,可以转卖、赠送、出租或者出借数字内容。

⑥ 可信执行:即在不安全的环境中保证程序按照预期的方式执行,程序的执行是安全可信的。

当前大部分 DRM 系统中,数字内容是经过加密、封装、添加水印和签名等处理后分发的,可以认为通过加密等处理的数字内容是安全的。此外,电子商务中的安全交易、电子支付等技术也是影响 DRM 发展的重要因素。

1.2 数字唯一对象标识

1.2.1 DOI 在 DRM 系统中的作用

在 DRM 系统中，每一个被保护的数字内容都有唯一的内容标识，它是鉴别不同数字内容的唯一参考对象。目前用于内容辨识的主要标准有 W3C 的统一资源标识符（Universal Resource Identifier, URI）、数字对象标识符（Digital Object Identifier, DOI）以及 MPEG-21 的数字项目辨识（Digital Item Identification, DII）等。在实际 DRM 系统中应用较多的是 DOI，它是由国际数字对象标识符基金会（International DOI Foundation, IDF）构造的一个框架，为数字环境中的数字对象分配唯一的、永久性的标识，方便该对象的管理和使用^[3]。

DOI 是针对数字资源的永久性标识符。DOI 实现了数字资源动态的持久链接，如果数字资源的 URL（Uniform Resource Locator，统一资源定位符）发生了变动，出版商只要向注册代理机构（Register Agent, RA）提交并更新数据即可保证链接的有效性；同时还提供一站式服务，即各出版商通过 DOI 系统实现引文到全文一站式的链接。目前，DOI 国际基金会拥有 8 个注册代理机构，上千万个已经分配并解析的 DOI 号码在美国、欧洲和澳大利亚以及非英语国家的各 DOI 代理注册机构注册，其应用范围已从科技领域拓展到了政府部门领域。

J. Dalziel 在《DRM 环境中的 DOI》^[4]中指出，DOI 是一个在数字环境中标识、交易知识产权的系统，它提供了一个管理知识产权内容、链接客户和内容提供商、方便电子交易以及自动化管理所有媒体的版权的框架。利用 DOI 可更容易、更方便地在网络环境中管理知识产权，构建电子商务的自动服务和交易，应用于 DRM 保护出版物的知识产权。

图 1-2 是基于 DOI 的 DRM 系统结构图，从图中我们可以看出，DOI 主要用于信息资源创建与保护以及信息资源发布这两个过程。在信息资源创建与保护的同时，把信息资源的 DOI、元数据及其 URL 向 RA 登记注册，存储这些信息，并由 RA 对其进行管理维护。发布信息资源时，信息资源的 DOI 信息与其一起发布，用户要获取数字资源或有关这一资源的相关信息时，DOI 查询请求就会被传送到 DOI 注册中心，由 DOI 解析系统即 Handle System（句柄系统）解析该 DOI 的 URL 地址，将其 URL 送回给用户浏览器并将结果显示给用户。通过 DOI 和 DRM 的结合，数字资源出版商或所有者可以对数字资源的内容设置权限，达到版权保护的目的。

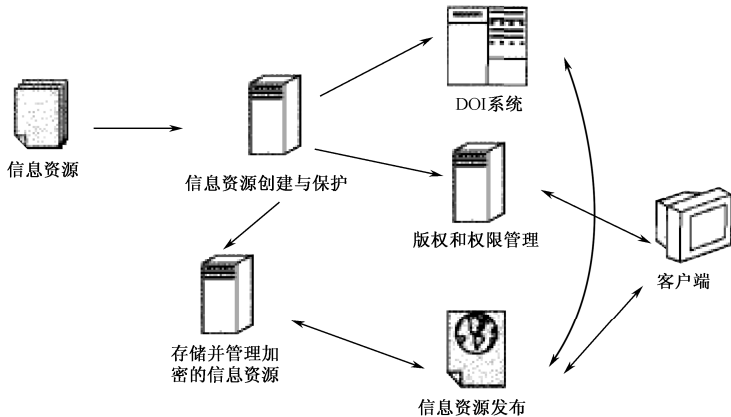


图 1-2 基于 DOI 的 DRM 系统结构图

1.2.2 DOI 系统概述

在任何数字环境中，唯一标识符对于信息管理都是非常重要的。如果不和分配者协商，在一个上下文中分配的标识符可能会在另外一个地方（或时间）遇到或者重用，分配者无法保证其他人知道他做的假设。

广义地说，DOI 系统是在数字网络环境下标识和交换知识产权对象的一种开放性系统，它遵从 URI 规范，并为基于数字对象结构公认标准的数字内容管理和数字版权管理提供了框架。这个框架是可扩展的，它有 4 个组成要素：标识符、解析系统、元数据和规则。狭义地说，DOI 是指标识任何数字化对象的一种标识符。DOI 实际上是一种 URI 或 URN（Uniform Resource Name，统一资源名称），即数字网络上的一个实体的名称（不是地址）。它为数字网络上的受控信息提供了一个持久可追溯的鉴别和可互操作的交互系统^[5]。

DOI 系统具有以下特点。

- ① 唯一性：如果被描述的资源被修改或者移动，其标识符并不需要改动。
- ② 互操作性：能够与其他来源的数据互相操作。
- ③ 可扩展性：通过对 DOI 管理可增加新的特征和服务。
- ④ 多元解析：能够针对 DOI 的元数据进行数字内容的多版本、多格式、多镜像解析服务。
- ⑤ 分布式服务和管理：能够为网络中的任何节点的 DOI 号码提供服务和管理。
- ⑥ 动态性：元数据、应用和服务的动态更新。

DOI 系统上述特征为用户提供了如下功能：用户可以知道自己拥有哪些资源，需要哪些资源，所需资源位于何处，如何获得所需资源，找到并使用那些存储位置发生变动的资源。

DOI 系统的构建使用了几个现存的基于标准的组件，这些组件组合在一起并进一步开发出了一个协调的系统。整个系统已经为 ISO（ISO TC46/SC49）作为标准所接受。DOI 已经发展为一个由 IDF 管理的跨产业、跨区域、非营利的成果，DOI 广泛应用于科技出版、政府文档、数据等众多领域，DOI 可能用来为数据鉴别提供可互用的通用系统。

1.2.3 DOI 的语法

DOI 标识符是一个唯一的编号，遵循 ANSI/NISO Z39.84-2000 的句法标准，是基于 Handle System 建立的。DOI 由前缀和后缀两部分组成，并用字符“/”分开。其格式为：

```
<DIR><REG>/<DSS>
```

DOI 前缀包括由“.”分隔的两部分。其中<DIR>为目录代码，由 Handle System 赋予固定值 10，这是为了区分其他使用 Handle System 技术的系统；<REG>为登记机构代码，主要是一些大型的出版机构，由 IDF 负责分配号码，一般为四位阿拉伯数字。

<DSS>为 DOI 后缀，由委托命名机构给定。要扩大 DOI 的影响和应用范围，就必须兼顾原有的传统的标识符系统，所以主要编码格式基于一些已存在的标准知识体系，如：期刊及文献内容标识符（Serial Item and Contribution Identifier, SICI）、出版物件标识符（Publisher Item Identifier, PII）、国际标准图书编码（International Standard Book Number, ISBN）、国际标准期刊编码（International Standard Serial Number, ISSN）等，避免破坏已经建立在上述标

识符基础上的商业关系。因此，允许使用者将传统的标识符作为 DOI 的组成部分沿用下来，同时也推荐这么做，这样就两个系统来精确确定了同一个实体。以下给出的一些例子都是合法的。

例 1：标识一本图书，后缀是基于 ISBN 编码，如 10.1008/ISBN0764548891。

例 2：标识 “the Digital Object Identifier System home page on the Web”，其前缀是 “10.1000”，后缀是 “1”，即标识符为 10.1000/1。

1.2.4 DOI 的解析

DOI 的解析是一个过程，即用一个标识符作为发送给网络服务的输入（请求），作为特定的输出返回和确定实体相关的当前的一个或多个信息（或状态数据），比如，可以找到目标的一个地址（如 URL）。解析在标识符和输出之间提供了一个受控的间接层。解析组件允许在 TCP/IP 网络上从 DOI 到关联数据的重定向。

DOI 的解析原理类似于 DNS 系统，在一个数字对象发布前，出版商赋予其一个唯一和永久的标识符号，并将标识符和数字对象的地址信息登入 DOI 系统。当用户查询某一 DOI 标识的数字对象时，系统就在其数据库中查找该 DOI，然后将与 DOI 相关的数字对象的信息返回。如果数字对象变更名称或移动位置，其所有权人负有更新数据的责任，以确保解析结果的正确。DOI 解析使用的是 Handle System 技术，它们为 DOI 提供了从单一解析（Simple Resolution）到多元解析（Multiple Resolution）的应用。Handle System 是由美国研究创新联合会（Corporation for National Research Initiatives, CNRI）开发的技术平台，用于因特网信息的命名、解析和管理，它与 DOI 的特点相吻合，提供高效、准确的解析服务。

DOI 主要的解析的方式如下。

1. 单一解析

单一地址解析机制为用户提供了对数字内容的唯一访问，即一个 DOI 号码对应一个数字内容。DOI 系统要有效地管理地址，出版商在为其每项数字内容注册 DOI 号码时，要同时向 Handle System 提交相应的 DOI 号码和网址（URL），并存放在 DOI Directory 中。出版商负责对 DOI 数据的维护，当资源地址发生改变时，出版商应通知 Handle System 做相应更新，以确保链接的有效性。

2. 多元解析

DOI 的多元解析机制为用户提供了更多的选择和便利。同样的数字内容在网络中可以有多种格式、多个版本存档，而且有可能存放在多个镜像站点，此外与其相关的数字信息都可以由 DOI 多元解析进行处理。例如，用户可以选择最近的镜像站点下载数据；可以链接到与查询数字内容相关的资源，如评论、主题作品、版权人及出版商的信息与联系方式等，从而达到对资源的深度利用。

DOI 主要是由出版界发起建立的一个标准，因此它考虑更多的是在出版界开展电子商务，保护知识产权和出版商的利益。同时，IDF 共有 6 家 RA 提供 DOI 注册、管理和应用服务，但是这些服务都不是免费的，除了注册的时候要交纳 1000 美元的费用外，每年还要向申请机

构交纳不菲的使用、维护费用，这对一般的中小出版机构是个不小的开支。同时，由于网络中使用最多的数字资源已经不再是电子书籍了，而是演变成了多媒体数字资源，现在无论国际还是国内都没有出现较为成熟的针对多媒体资源的数字资源标识符。

美国出版商协会（Association of American Publishers, AAP）设计了一个 DOI 登记系统，该系统将 DOI 唯一标识符与元数据信息、数据对象地址联系起来，集中保存在一个服务器上以供查找。与 URL 相比，这种 DOI 登记系统的好处是：URL 可能是不稳定的，同一个 URL 所对应的对象内容可能发生变化，同一对象的 URL 也可能改变、删除等；而 DOI 登记系统则保证了未来网络出版环境下进行数字内容管理的一个可能框架。

1.3 DRM 系统中数字内容的使用控制

DRM 系统对数字内容进行使用控制，确保只有授权使用者才可以使用受保护的数字内容。同时，用户对数字内容只拥有授予的使用权利，根据使用权利对数字内容进行访问。

在用户获得数字内容及其许可证后，如何按照授予的使用权利，控制数字内容的使用是 DRM 的关键。使用控制包括用户控制和权利控制：用户控制对数字内容的使用者进行控制，确保只有被授权者才能使用相应的数字内容；权利控制对数字内容的使用进行控制，用户只能在数字许可证描述的权利范围内使用数字内容，不能进行超出这些权利以外的操作。

1.3.1 用户控制

用户控制是确保数字内容合法使用，防止非法复制和非法共享的关键。通常有两种用户控制方式^[1]：基于额外专用设备的方式和基于身份标识绑定的方式。

1. 基于额外专用设备的方式

将用于控制数字内容使用的敏感信息（如数字内容解密密钥、数字许可证或者其他相关密钥等信息）存储在额外的专用安全设备（如 CM-Stick、Smart-Card 等）里，使得数字内容只能在带有该专用设备的机器上使用，防止非法复制和共享。

这种方式能有效地控制数字内容的使用，常用于专业性较强的计算机软件的保护。但是，额外设备往往需要额外开销，也会由于损坏或遗失带来维护更新等问题。另一个不容忽视的问题就是多个系统往往需要多个专用设备。因此，从可操作性、易用性和成本等各方面来说，该方式不适用于电子书、音乐、电影等大众化数字内容的版权保护。

2. 基于身份标识绑定的方式

利用用户身份标识信息加密敏感信息或者将用户身份标识和敏感信息关联，使得具有该身份标识的用户才能使用受保护的数字内容。当前，大部分 DRM 系统都采用这种用户控制方式。该方式必须在用户身份标识唯一的前提下，才能有效地防止非法复制和共享。

目前，存在以下两类身份标识方法。

（1）用户标识法

使用用户信息标识用户，既可以是用户姓名、口令、E-mail 地址、电话号码、信用卡号、社会保险号等信息，也可以是用户指纹、视网膜等生物信息，还可以是可信第三方提供的用户数字证书信息等。

除了使用数字证书作为标识这种方式以外，基于用户标识的方式的好处在于可以实现多机访问。用户标识绑定方式存在以下问题。

① 用户姓名、口令、E-mail 地址、电话号码很容易在用户之间传递，很难控制非法共享行为。解决该问题的一个方法就是进行在线控制，确保每个用户每次只能在一台设备上使用购买的数字内容。但这要求用户在线使用数字内容，用户购买和使用数字内容的隐私容易被跟踪，且可能增加上网费用。

② 使用信用卡号和社会保险号标识用户存在严重的隐私和安全隐患，用户难以接受；

③ 使用用户指纹、视网膜等生物信息标识用户需要额外设备，实现起来不方便；

④ 使用数字证书标识用户会对数字内容使用速度造成影响，且在证书更新后，原先下载的相关数字内容或数字许可证也必须更新。

（2）设备标识法

使用设备信息标识用户，可以是 IP 地址，也可以是硬件标识（如计算机的 CPU 序列号、硬盘识别号等）。Microsoft 在 Windows XP 软件激活技术中，采用了 10 个硬件标识信息作为用户身份标识。基于设备标识的方式使得数字内容只能在相应的设备上使用，能有效防止非法复制和共享。

由于 IP 地址动态分配的情况普遍存在，基于硬件标识绑定的方式更为常见。但是，该方式存在以下几个问题。

① 用户隐私问题，存在购买隐私容易被跟踪的隐患，这是基于身份标识绑定的 DRM 系统普遍存在的问题。通过硬件标识随机化机制，使得用户每次购买数字内容时，提交的硬件标识信息都是不同的，从而降低了购买隐私被跟踪的可能性。

② 设备更换问题，用户更换设备后，无法使用原先购买的数字内容。解决该问题的关键在于实现权利迁移，这方面已经有不少研究成果。

③ 硬件变更问题，当一个或几个相关硬件发生变化后，原先购买的数字内容就无法使用了，即使这种变化很小。Macromedia 通过重新获取许可证的方式来解决这个问题，但这需要用户重新连接版权服务器，设备硬件配置信息将被跟踪记录，存在隐私问题，还会增加服务器负担。WinXP 和 eCX 通过在用户本地保存原先硬件配置信息、比较新旧硬件配置信息的方式，在一定范围内自适应硬件的变更。但这种将原先硬件配置信息保存在用户本地的方式存在安全隐患，新旧信息的比较点很可能成为攻击点。

综上所述，基于额外专用设备的用户控制方式对保护计算机软件等高价值信息来说是有效的。而对于一般大众化的数字内容而言，基于硬件标识绑定的用户控制方式更为合适。现有的用户控制机制还有待改进，特别需要加强考虑用户的需求，提高用户对 DRM 系统的接受度。

1.3.2 权利描述与控制

数字内容的使用权利由权利描述语言（Rights Expression Language, REL）描述，目前主要的权利描述语言有 XrML、ODRL、MPEG-21 REL、LicenseScript 等。REL 是 DRM 领域的重要研究课题，主要用来向权利提供方规范和描述用户对数字内容的授权使用策略，其中主要包括使用权限、条件和约束规则。

权利控制的关键在于权利解析和验证。权利解析和验证是解析数字许可证，验证用户的操作是否在许可范围内，前提条件和限制条件是否都已经满足。权利解析和验证与所采用的权利描述语言密切相关，由 REL 解析器处理。REL 解析器一般内嵌于 DRM 控制器中。

XrML（Extensible Right Markup Language，可扩展版权标记语言）是一种基于 XML（Extensible Markup Language，可扩展标记语言）的权利描述语言，在 DRM 系统中它可以以许可证的形式描述数字内容的访问政策。XrML 许可证定义谁可以访问内容，内容如何受到保护和分发，控制详细的使用权（如授权打印和基于时间的权限），执行具体操作。

图 1-3 给出了一个基于 XrML 的权利解析和验证流程实例。图中，用户要求播放一个受保护的音频文件。收到用户的播放请求后，播放控制程序——XrML 启动应用程序（XrML-enabled Application）收集用户信息、文件信息和用户操作信息，向 REL 解析器——许可证解释器（License Interpreter）发出请求。解析器获取对应的数字许可证文件并验证其有效性，确认该文件未被篡改，然后解析许可证文件，检查是否包含播放权限，并验证该权限是否确实为当前用户所拥有。如果检验通过，则返回相应的限制条件和前提条件给播放控制程序。播放控制程序要求条件验证器（Condition Validator）验证是否满足相关条件（如播放次数是否在许可范围内等）。如果条件满足，则用户可以播放该音频文件^[1]。

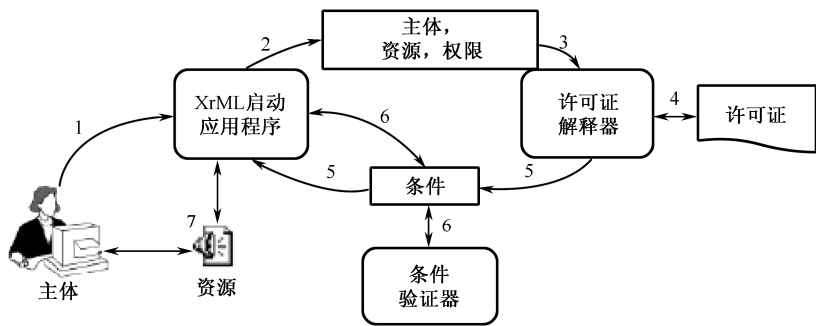


图 1-3 权利解析和验证流程

作为权利描述语言，语义的无二义性是本质问题。如何基于 REL 正确地描述被购买内容的使用权利，并确保权利之间无冲突，是值得我们考虑的问题。近年来 REL 研究工作也集中在形式化的 REL 描述。

表 1-1^[6]给出了代表性的 REL 及形式化使用控制模型在数字权利使用控制中典型特征的比较。符号○、×和—分别表示具备、不具备或不适合某特征。

表 1-1 数字权利描述语言与形式化模型的使用控制特性比较

数字权利 使用控制	规范的数字权利描述语言				形式化的 REL 与使用控制模型			
	XrML	ODRL	OMA	MPEG-21	LicenseScript	LiREL	UCON _{ABC}	UCON _D
“否定” 权利	—	○	—	—	—	×	×	×
约束与 义务	○	○	×	×	○	○	○	○
版权实现	×	×	×	—	○	×	—	×
权利管理	×	×	×	×	○	—	○	○
形式化 方法	○	○	—	集合 描述	多重集重写规则 Prolog 逻辑	集合 描述	集合描述 一阶逻辑	集合描述 一阶逻辑
权利可 转移	○	○	×	○	○	○	×	○

现有的 REL 针对一些具体的 DRM 应用场景，不能合理有效地描述权利管理。随着商业模型的扩充和新的数字权利的提出，一些特征被不断引入 REL，造成了 REL 核心语义不断扩展，以支持新的商业模型中的权利管理需求。这个问题的出现主要是由于缺少权利管理和权利描述的分离，导致 REL 变得复杂、难于理解和操作。为此，在层次性 DRM 系统^[5]中提出了一种基于核心语义描述和权利管理分离的 DRM 服务框架。该框架具有两个优势：一是在不改变内核语言的基础上通过引入新的协议来实现扩展权利管理的能力；二是用户的终端设备仅仅需要支持简单的核心语义，而复杂的权利管理功能被放置在服务器端处理。然而，作为一个总体概念模型，它并未考虑授权使用控制中的可信性问题；其次，针对数字权利管理、许可分享、用户认证等关键技术，仍缺少具体的实现机制和安全协议^[6]。

值得注意的是，最近由有关 REL 和 DRM 权利管理的研究认为，数字权利使用也可以被看成内容分发、传播和使用过程中一种特殊的、持续的访问控制机制。这明显不同于传统的访问控制方法。因为传统的访问控制主要集中在实体授权和资源访问前的合法权利判定，例如自主访问控制（DAC）、强制访问控制（MAC）和基于角色的访问控制（RBAC）三种重要的访问控制策略及相关模型。

Usage Control（简称 UCON）是一种可用于 DRM 应用的基础访问控制框架^[7]，它融合了授权（Authorization）、义务（oBligation）和条件（Condition）三个基本组件，也被称为 UCON_{ABC}。该框架具有持续的访问控制特性，并且易于描述资源使用过程中实体属性的动态变化。在 UCON_{ABC} 中，属性的变化通常体现在权利实施前、后以及作用过程中，同时结合三个基本组件，构成了 UCON 模型家族（图 1-4）。Pretschner 给出了使用控制在可用性、实现及非功能性质等三方面的系统分类^[8]。文献[9]中提到权利管理和内容保护成了为目前 DRM 系统的关键脆弱点，为此提出了一个四层安全模型，包括信任层、权利管理层、权利实施层和内容保护层等。Nair 和 Tanenbaum 等人基于 UCON_{ABC} 提出了一个支持 DRM 的 Trishul-UCON 框架，并实现了它基于 JVM 中间件的跨应用 DRM 策略^[10]。

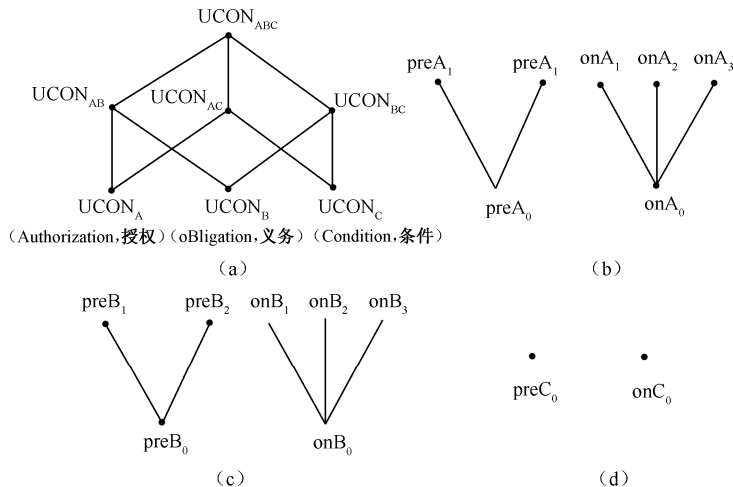


图 1-4 UCON模型家族

图 1-4 显示了 $UCON_{ABC}$ 的组合, 以及它们之间的关系。其中 A、B、C 被看做地位同等的, 因此这三个基本模型在底部; 第二层是它们之间的两两组合; 最上层是包含所有三个的组合。用这种方法可以简洁地表示 A、B、C 在给定上下文中的组合。A、B、C 模型可以分别划分成由图 1-4 (b)、(c) 和 (d) 表示的几种情况。图 1-4 说明了 $UCON_{ABC}$ 模型空间的丰富性。

1.4 权利转移

合理地分享所购买的数字内容及数字权利, 有助于一个完整的 DRM 系统及其价值链的延伸和扩展。DRM 支持数字内容使用权利的转移, 可以转移到另外一台设备上, 也可以暂时或永久地转移给其他用户, 使得用户可以更换数字内容使用设备, 可以转卖、赠送、出租或者出借数字内容。

权利转移包括权利在设备间的转移和用户间的转移, 前者为权利的迁移 (Migration), 后者为权利的二次分发 (Redisrtibution)^[2]。权利可以迁移到另外一台设备上的特性, 称为可移动性 (Portability)。

支持权利转移在 DRM 研究中具有极其重要的意义, 它一方面可以提高用户购买、使用数字内容的积极性, 增强用户对 DRM 系统的接受度; 另一方面可以减少用户破解 DRM 系统的动机。

1.4.1 权利迁移与共享

目前, 通常有以下几种权利迁移方式。

① 基于额外专用设备的方式: 数字内容能且只能在带有该专用设备的机器上使用, 通过专用设备的可移动性来实现权利的可移动性。

② 基于用户绑定的方式: 主要包括基于 Rights Locker 架构的方式和基于用户关键信息绑定的方式。前者通过中心服务器的在线控制, 实现数字内容的移动使用。后者通过用户信用卡号等关键信息绑定数字内容, 实现权利的可移动性。

③ 基于 check in/check out 的方式: 该方式结合硬件标识绑定技术和 Rights Locker 架构技术的特点, 由中心服务器管理权利, 并通过设备硬件标识绑定的方式控制数字内容的离线使用, 实现权利在设备间的转移。以 SealedMedia 公司的 DRM 系统为例, 具体做法是: 用户首次申请数字许可证时, 版权服务器生成并保存漫游许可证, 然后生成与当前设备绑定的数字许可证副本发送给用户。此时, 数字许可证就被 check out 到指定的设备上。当数字许可证处于 check out 状态且 check out 期未届满时, 该许可证不可以再被 check out, 而当数字许可证被 check in 时, 当前设备保存的许可证副本将被销毁, 其他的设备就可以 check out 该许可证了。

④ 基于小范围共享的方式: 允许少量设备共享数字内容, 支持数字内容在不同设备上的使用。对同一个数字内容而言, 只要获取数字许可证的设备数在限定的范围内, 用户就可以通过新的设备再次从版权服务器获取许可证, 使用数字内容。如 Microsoft 允许用户一年内能够在四台不同的计算机上激活 Windows XP 操作系统。

⑤ 基于备份/还原 (Backup/Restore) 的方式: 通过备份数字许可证并将备份还原到其他设备的方式, 实现权利迁移。

上述 5 种方式中, 前两种是实现权利迁移的最直接有效的方法, 特别是 Rights Locker 架构技术, 除了可以实现权利在设备间的转移外, 还可以很好地实现权利的备份、二次分发等。但是, 对一般数字内容而言, 这两种方法存在不少问题。后 3 种方式都是针对基于硬件标识绑定的 DRM 系统提出的, 存在以下的特点和问题。

① 第 3 种方式与 Rights Locker 架构方式相比, 减轻了对网络连接的要求, 但仍需要服务器保存用户购买的数字内容权利信息并在线进行 check in/check out 操作, 存在易用性和隐私问题。此外, check in/check out 的状态需要在客户端保存, 非法用户很有可能通过硬盘克隆方式, 达到非法共享数字内容的目的。此安全问题是当前 DRM 领域的一大难点问题。

② 较第 2、3 种方式而言, 第 4 种共享方式能更好地保护用户的隐私。但数字内容只能在限定数量的设备上使用, 没有真正实现权利的可移动性。

③ 第 5 种方式理想的做法是权利在设备间转移时, 不需要额外的网络连接, 备份和还原都通过客户端 DRM 控制程序进行。其难点在于如何保证备份被还原后, 除目标设备以外, 其余的设备都不能使用该数字内容, 或者如何保证同一备份不能被无限次地还原到不同的设备。目前, 解决该问题的一个方法就是在线进行还原操作, 由中心服务器控制备份的还原次数, 如 Microsoft WMRM, 但又带来了用户隐私问题。更为主要的是, 与第 3 种方式类似, 非法用户能够通过硬盘克隆方式, 在源设备恢复已迁移的许可。

权利迁移的关键在于实现权利从一台设备到另一台设备的转移。对基于硬件绑定的 DRM 系统而言, 现有的权利迁移方案在有效性和用户隐私保护方面还存在不少问题。

OMA RI (Rights Issuer, 权限对象发行者) 在为用户许可授权时, 通常采用内容-许可-设备绑定的方式, 使得数字内容的共享使用受到了较强的控制。Digital Video Broadcasting 联盟最初为了便于内容在不同设备上的共享使用, 首先提出了“授权域 (Authorized Domain)”概念, 随后 OMA DRM 方案也在多个版本中使用了这一概念, 并实现了 RI 对域的统一管理, 包括创建和撤销域、用户设备的加入与退出域等, 域内设备之间可以共享内容和数字权利。但由此增加了 RI 的负担, 并成为了授权域的瓶颈; 此后, OMA DRM 在后续版本中通过引入域管理器对此进行了改进。目前, DRM 数字内容共享研究侧重于家庭网络域和个人娱乐域 (Personal Entertainment Domain)。文献[11]给出了一个 DRM 授权域的安全架构及其安全协议,

但它不支持 RO (Rights Object, 权限对象) 的转移和内容共享。文献[12]在家庭网络域 DRM 中进行了改进, 在引入本地域管理器 (Local Domain Management) 的基础上, 代替 RI 实施域成员设备的许可证分发和共享, 同时通过委托 RO (Delegated RO) 和代理证书 (Proxy Certificate) 实现了数字权利委托与共享。然而 LDS 的引入增加了系统开销和被攻击的对象, 并且数字内容的共享机制仅限于家庭网络。如何将其推广至广域网络, 需要做进一步的考虑^[6]。

Barhoush 等人^[13]提出了数字内容安全多播的 11 项安全需求, 并针对这些需求, 详细分析了现有代表性的 DRM 商用系统所存在的相关特征和不足, 以及改进的方向。文献[14]为改进现有的数字权利分发过程中许可配置受限的问题, 给出了一个 OPA (Onion Policy Administration) DRM 模型, 该模型使内容创建者和分发者都可以配置许可, 并且具有可追踪性, 有效地提高了数字权利分享的效率和安全性。Bhatt 等人^[15]在 Motorola E680i 智能移动终端上实现了个人化 DRM (Personal DRM) 原型系统, 使终端用户可自主地设置数字许可 (License), 并且灵活地在设备间转移许可, 达到了保护个人数字内容的目标。此外, Lee 提出了一个基于设备间内容使用时间的二次分发方案及其安全协议, 它是对数字权利中时间要素分享的一次有益尝试, 拓宽了数字权利分享研究的视野^[16]。冯雪和汤帆等人提出了一个 DRM 许可分享方案, 该方案基于遍历加密 (Ergodic Encryption) 和机器认证技术的许可获取与分享机制, 减轻了传统依赖授权域的成本负担^[17]。

1.4.2 二次分发

二次分发同时涉及权利在不同设备、不同用户之间的转移, 是所有权的转移, 比迁移更加复杂。其难点在于如何确保所有权的有效转移, 如何确保源用户暂时或永久丧失已转移的数字内容使用权利, 目标用户如何暂时或永久拥有相应的数字内容使用权利。

有以下两种不同的二次分发模式^{[1][18]}。

1. 基于本地 DRM 服务中心 (Local DRM Service Center, LSC) 的二次分发模式

通过在客户端引入 DRM 服务中心的方式来实现二次分发。LSC 类似于外部 DRM 服务中心 (External DRM Service Center, ESC, 如图 1-3 所示的许可证服务器), 用于存放对等用户 (即二次分发目标用户) 的信息、认证对等用户、生成对等许可证 (Peer License, 包括对等二次分发许可证 Peer-Redistribution License 和对等租借许可证 Peer-Loan License)。只有具有正式许可证 (由 ESC 生成) 或者对等二次分发许可证的用户, 才能进行二次分发。图 1-5 给出该模式下的二次分发流程, 图中用户 C 将数字内容 (音乐) 转让给用户 D (权利的永久转移), 用户 D 将数字内容出借给用户 E (权利的暂时转移)。

2. 基于水印证书的二次分发模式

通过水印证书认证机构 (Watermark Certification Authority, WCA) 颁发的水印证书, 识别非法分发数字内容的用户, 通过数字内容发行者 (Content Distributor, CD) 实现二次分发。图 1-6 给出了该模式下的二次分发流程。图中, Alice 将她从 CD 处购得的数字内容 X 转让给 Bob。其中, K_B , $Cert_B$ 分别是 Bob 的公钥和公钥证书, W 是 Bob 的水印, $WCert_B$ 是 Bob 的

水印证书, 包含 $(E_{K_B}(W), \text{Cert}_B)$ 和 $\text{Sign}(E_{K_B}(W), \text{Cert}_B)$ 。 $W\text{Cert}_A$ 是 Alice 的水印证书, V 是 Alice 从 CD 购买 X 的交易标识, V' 是此次二次分发的交易标识, $X' = X \oplus V'$ 是将 V' 作为水印插入 X 后的结果, δ 是一个 m 元随机置换。这里, \oplus 是插入水印运算, E 表示加密, $\text{Sign}(\cdot)$ 是 WCA 的签名。

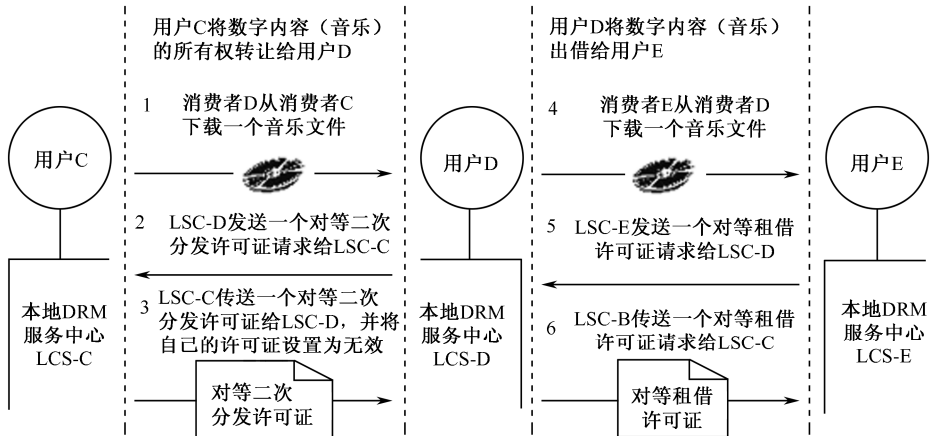


图 1-5 基于LSC的二次分发流程

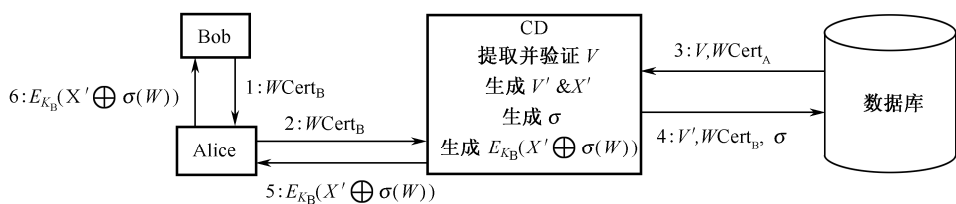


图 1-6 基于水印证书的二次分发流程

上述两种二次分发模式中，第一种在现有 DRM 架构的基础上，引入 LSC 来实现二次分发。该模式只涉及分发者和接收者，不需要任何第三方的参与。但是，该模式增加了客户端程序的复杂性和运行负担，同时这种在用户本地处理二次分发、生成对等许可证的方式，对 PC 等通用设备而言，存在安全隐患。此外，分发者可以通过硬盘克隆等方式，恢复已转移的数字内容使用权利，并可以再次进行转移。

第二种模式在每次二次分发过程中，都需要向 CD 发送数字内容副本，由 CD 进行插入水印、公钥加密等处理，服务器负担重、花费时间长，且不适合超级分发（Superdistribution）等商业模式。该模式需要可信第三方 CD 的参与，带来了用户隐私问题。更主要的是，源用户转让数字内容后，虽然不能再次进行转让，但仍可以使用该数字内容。而且，只能通过 CD 从用户本地的内容副本中提取交易标识水印信息的方式来发现这种非法行为，难度较大。

上面的二次分发模式未能有效地解决所有权转移问题。参照权利迁移方式，采用基于额外专用设备或者 Rights Locker 架构的方式能较好地解决该问题。但这又带来了成本、用户隐私、易用性等各方面问题。

权利转移问题是 DRM 领域的一大难点问题。实现权利有效转移的关键在于权利的监控和跟踪。基于 Rights Locker 架构的 DRM 系统在这方面具有很大的优势。针对基于硬件绑定

的 DRM 系统的特点,引入许可激活机制和许可检测机制,通过中心服务器的监控和跟踪,使得一个数字内容只能转移到一台设备上,非法恢复已转移许可的设备无法再次申请和转移数字许可证,孤立和隔离非法设备,保护数字内容的版权。这种方法在一定程度上提高了权利转移的有效性,但仍需要可信第三方的参与。目前,就 PC 等通用设备而言,单纯的双方交互很难确保权利的真正转移。借助安全硬件模块的 TCG 和 NGSCB 技术将会为权利转移提供一种新的思路。此外,部分权利转移问题将会是另一个新的研究点^[1]。

1.5 可信执行

1.5.1 DRM 的安全性

传统计算机在体系结构上存在许多漏洞,缺乏很好的硬件防护措施。现有的大部分 DRM 系统中,数字内容的解密使用、使用权利的解析验证由客户端 DRM 应用程序负责。对 PC 等通用设备而言,客户端 DRM 应用程序的运行环境是不安全的,必须采取一定的防篡改机制,保证 DRM 应用程序的安全性,确保数字内容的合法使用。根据是否使用专门的安全硬件设备,目前主要有以下两类防篡改机制。

1. 基于软件技术的防篡改机制

其主要包括篡改检验机制、代码加密机制和代码模糊机制。篡改检验机制采用校验和、断言校验等技术检验软件代码的完整性,一旦发现任何非法修改,便终止程序的部分或全部功能;代码加密机制通过加密部分软件代码的方式,防止恶意用户跟踪、剖析和修改程序代码;代码模糊机制通过对软件代码的词法、数据结构、程序的控制流程和逻辑的模糊化,使得程序不易理解和分析,防止黑客的恶意修改和破坏。此外,基于软件技术的防篡改机制还需要配合各种反跟踪技术,以提高防篡改机制本身的安全性。

2. 基于硬件的防篡改机制

借助安全的硬件设备,保证程序在可信环境中执行,防止非法程序访问受保护的数字内容。这里,安全的硬件设备是一个包含 CPU、存储器的防篡改独立处理引擎,具备安全存储数据、逻辑处理、数据加解密等能力,能够抵抗各种形式的软件攻击和一定的硬件攻击,可以是专用的安全外围设备,如 Smart-Card 等,也可以是集成到硬件平台的可信处理模块,如 SPU (Secure Processing Unit), TPM (Trusted Platform Module), SSC (Security Support Component) 等。

上述两类防篡改机制中,前者采用软件技术手段来增加恶意用户剖析、修改、破坏程序源代码的难度,减少程序被破解的可能性;后者则通过专用的安全硬件设备提供的可信空间,保证相关程序的安全运行,防止外部非法程序的攻击。第二类防篡改机制更加强健,能更好地抵抗各种形式的软件攻击。现有的大部分 DRM 系统都采用第一种防篡改机制来保护 DRM 应用程序。另外,采用升级机制,不断修补系统以增强系统的安全性,也是经常采用的方法。

最近几年，对计算机安全和可靠性的依赖促进了可信计算（Trusted Computing, TC）技术的发展。1999 年 Compaq、HP、Microsoft、IBM 和 Intel 成立了可信计算平台联盟（Trusted Computing Platform Alliance, TCPA），TCPA 提出了可信计算的思想用以保护计算终端的安全。其主要思路是基于安全硬件和安全操作系统来实现一个可信的平台，并将信任延伸到客户端、服务器、网络和通信平台。TCPA 于 2003 年更名为可信计算组织（Trusted Computing Group, TCG）。TCG 为了向应用软件及平台间提供信任保证，定义了一组提供基于硬件的信任根的标准和一组基本功能。TCG 中的信任根是指主板上的被称为 TPM（Trusted Platform Module，可信平台模块）的硬件组件。TPM 通过与 TPM 永不分离的根密钥来保护数据，并提供基本加密功能，如随机数生成、RSA 密钥生成和 RSA 非对称密钥算法。更重要的是，TPM 提供了平台完整性的度量、存储和报告，进而提供了严格的保护和证明。

TCG/NGSCB 技术借助安全的硬件模块 TPM/SSC，以密码技术为支持、安全操作系统为核心，通过硬件和软件的结合，建立硬件安全模块支持下的可信计算平台，通过平台证明（Platform Attestation）功能，确保硬件环境配置、操作系统内核、服务及应用程序的完整性，保证用户工作空间的完整性和可用性，确保数据存储、处理、传输的机密性和完整性，保护系统免受恶意代码的攻击，确保系统按照预期方式执行。

系统安全性问题是 DRM 领域的一个重要问题，用户终端的安全平台是保障数字内容按照许可证所描述的数字权利安全、可控、可信赖执行的基础。随着可信计算技术的发展，人们开始探讨利用它来保证 DRM 系统的安全性。

1.5.2 可信平台模块

可信平台模块（Trusted Platform Module, TPM）是一块嵌入 PC 主板的系统级安全芯片，它集成了数字签名、身份认证、信息加密、内部资源的授权访问、信任链的建立和完整性测量、直接匿名访问机制、证书和密钥管理等一系列可信计算所必需的基础模块，为各种安全应用提供了一个功能强大的平台。

TPM 作为可信计算平台的核心安全控制和运算部件，必须支持一套最小化的算法和操作，用以提供平台身份验证，数据安全存储和平台完整性校验等功能^[19]。因此，TPM 必须具备四个基本技术特征：内存屏蔽（Memory Curtaining）、安全输入输出（Secure I/O）、平台身份的远程证明（Remote Attestation）和封装存储（Sealed Storage）。

1. 内存屏蔽

内存屏蔽是指通过一种强健的、硬件增强型的内存隔离特性，避免程序间相互读写内存中内容。

2. 安全输入/输出

某些入侵者通过按键记录器或屏幕捕捉器来收集用户计算机上的信息，如用户 ID、口令、文档及电子邮件的内容。安全 I/O 提供了一个从键盘到程序再由程序到屏幕的安全路径。

3. 远程证明

平台可以利用完整性度量 and 存储来产生一个完整性报告，并通过一种称为“证明”的挑战-响应机制传递给其他平台。

4. 封装存储

TCG 的目的在于为秘密信息，如加密密钥等提供可靠的、基于硬件的保护。由于开放的计算机平台可以运行任何软件，这种目的在于确保被保护的秘密信息只有在平台的软件状态完全符合已定义的度量标准时才可使用。TCG 的封装存储特性可用于将一个受保护的秘密信息和特定的软件配置绑定。如果配置值不符合定义的标准，封装的密钥将不会被释放，从而避免密钥遭到病毒破坏或被入侵者利用。

TPM 作为可信计算平台的核心安全控制和运算部件，它的工作要先于操作系统和 BIOS，不可能使用计算机的内存和外存，因此必须内部实现一些公开的安全算法，以便于其部件的接口标准化，和提供一些安全操作的密码运算。所以，一个 TPM 产品至少需要内部实现一个基本的密码算法集合，根据 TCPA 的技术规范，这个集合至少应该包括 RSA、SHA-1、HMAC 三种算法，也可以包含更多的算法，如 DSA 或 ECC。其中 RSA 主要用于加密和签名认证，SHA-1 和 HMAC 是两种生成报文摘要的算法。

总体来讲，TPM 的体系结构中包括的部件如图 1-7 所示。

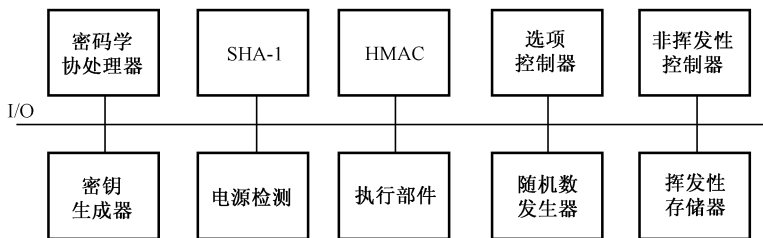


图 1-7 TPM 体系结构

1.5.3 可信平台的信任链度量机制

首先介绍可信度量根（Core Root of Trust for Measurement, CRTM）的概念。CRTM 是一种可度量的信任根，即信任链的源头。一般的形式是 BIOS 之前的运行代码或者是具备此功能的 BIOS 本身。

解决平台安全问题的核心办法之一就是建立一个可信的信任链。在 TCG 系统中，由于信任根的错误行为不会被检测到，因此信任根必须是一个可信的组件。在一个可信计算系统平台上有两个根：完整性度量信任根和完整性报告信任根。图 1-8 是可信平台的信任链度量机制。

这个信任的传递过程是通过完整性度量及其验证来实现的。控制首先传递给 TPM 以检查 CRTM 的完整性。然后 CRTM 计算 BIOS 代码的散列值，并以一种历史记录可测量的方式将此值存储在 TPM 的寄存器 PCR 中。PCR 在一个引导期间不能被删除或改写，只能与其他一些值连接后对此值进行更新。计算散列值后，将此值与已经存储的 BIOS 的散列值进行比较，

如果一致，则 CRTM 将控制传给 BIOS 代码。随后执行 BIOS 代码。BIOS 通过 TPM 对系统组件、外围设备选项 ROM 进行测量和计算，并将相应的值存储到 PCR 中。这些值在 OS 加载程序中将会被读取出来并与 OS 加载程序自己计算的值进行比较。如果一致，则控制传递给 OS 加载程序。OS 加载程序对 OS 及带各种应用的 OS 进行同样的操作。如果代码在任何阶段被修改了，则通过散列值便可检测到。否则用户认为代码没有受到攻击，可以将控制传给它。这种从 CRTM 到 BIOS，再从 OS 载入程序到 OS 的度量步骤，被称为信任链，即安全引导的过程。OS 在任何时候都可以利用 TPM 对其他应用进行度量。

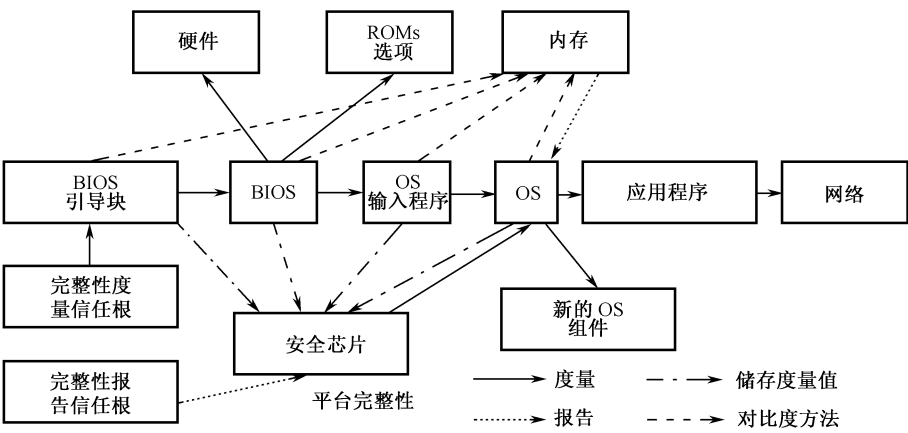


图 1-8 可信平台的信任链度量机制

1.5.4 数字权利的可信执行

近年来，学术界开始研究可信计算在 DRM 领域的基础应用。基于可信计算中的关键技术，如远程证明、Seal 技术等，以及完整的可信终端平台体系，研究 DRM 内容使用策略的可信分发、安全存储和 DRM Controller 可信执行。文献[20]综述性地探讨了可信计算的进展及其基本特征，并阐述了基于可信计算技术的可信移动 DRM 健壮性实现，主要包括设备密钥的安全存储、基于 Seal 的内容分发与访问等。文献[21]中较为详细地阐述了基于 TCG 规范的移动终端平台架构，指出了移动终端平台所需的基本 TPM 指令和函数，并从终端保护、移动代码安全等角度讨论了移动代码授权问题，以及基于远程证明的移动终端平台验证和 DRM 内容保护。文献[22]也给出了一个概念性的可信移动平台体系架构。文献[23]从 DRM Controller 平台环境的角度，探讨了现有的 OS 不能有效地支持可信计算中的远程证明和 Seal 技术，目前开放终端平台上的主流 OS 及其访问控制模型无法保护对解密后数字内容的直接访问和输出，并探讨了许可策略的可信实施，以及需要进一步构建基于虚拟机技术的隔离执行环境、实施参考监控器（Reference Monitor）概念并加强强制访问控制模型的实现等。

目前可信的移动终端平台环境和移动应用（Mobile Application）类数字内容的安全等领域也开始受到了关注。通过文献[24]中提出的一个基于 TPM 和具有隐私保护的 SITDRM 实现可信终端的方案看出，DRM 与可信计算技术是融合的、互补的。结合 Xen 虚拟技术及架构，通过支持第三方远程证明协议 AP2RA，可实现对用户终端平台关键部件（如 DRM 控制器等）

的远程验证, 确保数字权利的可信执行^[25]。

但是, 也有人批评 TCG/NGSCB 是一种过度保护的 DRM 方式, 其远端证明 (Remote Attestation) 功能将使软件提供商和数字内容提供商可能通过检验用户的机器是否是受信赖的机器来确定是否允许用户存取资料, 减少用户对硬件的控制, 使人们逐渐丧失对数字内容的公平使用权。此外, 还可能存在不公平竞争、侵犯用户隐私、市场垄断等问题。同时, 用户必须升级或更换现有机器才能使用 TCG/NGSCB 技术, 这使得将 TCG/NGSCB 技术广泛应用于一般商业领域在近期内是不大可能的^[1]。

1.6 互操作性

1.6.1 DRM 系统互操作的现状

目前已有许多商用的 DRM 产品, 如 WMDRM、Apple iTunes 的 FairPlay、Open Mobile Alliance 的 DRM 以及 Real Network 的 Helix DRM 系统等。然而, 绝大多数现有 DRM 系统的一个突出问题是: 当前 DRM 是一个封闭的系统, 没有统一的标准。此外, 各个 DRM 提供商为了保证各自的利益最大化, 它们的 DRM 系统往往只支持自己的文件格式 (如 Microsoft 的 ASF 格式、Real Network 的 Real 格式、MPEG-4 等)、编码格式 (如 MP3、WMA、ACC 等) 以及各自专有的多媒体保护方法。这些系统间彼此不兼容, 用户从某个 DRM 系统服务提供商购买的数字内容只能在支持该系统的数字设备中使用, 不能在支持其他 DRM 系统的数字设备中使用。例如, 用户购买了使用 iTunes FairPlay DRM 保护的数字多媒体内容, 就只能在 Apple 的设备上播放, 而无法在 Windows Media 播放器上使用。这样的后果是, 用户不得不购买同一数字内容的多个版本。这限制了用户的选择范围, 给他们在数字内容的使用上带来了极大的不便, 大大降低了人们使用 DRM 系统的热情, 也严重阻碍了数字出版市场的进一步发展。

随着网络影响的增大, 用户希望在任何时候任何地方使用任何设备在任何可行的商业模式中获得和使用任何内容。目前解决该问题最主要的手段是采用互操作技术, 通过某种技术手段使不同的 DRM 系统之间能够相互访问, 用户可以方便而透明地使用不同的 DRM 系统提供的内容服务。

DRM 互操作是指一个 DRM 系统具有使用其他 DRM 系统的一部分或其设备的能力, 是一种技术与另一种技术进行交互以实现相应功能的能力。DRM 互操作的本质是来自不同内容提供商的不同实体 (可能是 DRM 系统、应用或模块) 之间能相互操作或共同工作的能力, 此过程中需要涉及两个或两个以上的实体^[29~31]。DRM 互操作从提出开始就得到了研究领域的关注。

有研究者分析, 要实现 DRM 系统之间的互操作性可采用以下三种基本方法^[32~34]。

1. 全格式 (Full Format) 模式

在这种模式下, 数字内容都采用统一的标准和格式。对用户而言, 这种模式是最为有利

的，能够提供方便的共享方式。但是这需要所有的提供商对文件格式达成一个协议，目前市场上使用的 DRM 系统在商业模式等方面各有不同，要建立一个统一的并能兼顾到各参与方利益的标准非常困难，难以得到内容提供商的支持。

2. 连接（Connected）模式

这种模式通过第三方实体对数字对象进行转换，将一种 DRM 系统中的数字内容和许可证转换为能够在另外一种 DRM 系统中使用的数字内容和许可证，使其适应其他的 DRM 机制。这种模式下的 DRM 互操作目前有两种方式：P2P 连接模式和 Broker-based 连接模式。这是目前最好的 DRM 互操作策略。

3. 配置驱动（Configuration Driven）模式

这种模式通过从服务器端下载驱动来实现互操作，即用户需要使用其他 DRM 系统的数字内容时，可从相关服务器下载驱动工具，然后在本地使用该工具将原始内容转换为可以使用的内容。目前这种模式被 MPEG 采用。

DRM 系统的互操作是具有挑战性的课题，问题涵盖面广，并牵涉众多利益体，解决起来非常复杂。目前对于 DRM 互操作技术的研究方兴未艾，但是还没有一个适用于所有 DRM 系统的成熟标准。下面介绍几种典型的互操作解决方案。

1.6.2 Coral DRM 互操作框架

Coral 联盟（Coral Consortium）是一个跨行业的团体，其组成宗旨是提高市场中 DRM 系统之间的互操作性。在已经实现的 DRM 互操作解决方案中，Coral 架构^[35]是一个非常著名和有影响力的机制，它对现有原始 DRM 系统的改动最小。Coral DRM 互操作框架（以下简称 CorallIF）是层次结构连接模式的 DRM 互操作机制，在该框架栈的底部是 Coral 体系结构，可将其当做一个建立可互操作系统的工具来使用。

Coral 采用一种许可证推导（License Derivation）的方式实现 DRM 互操作，新的许可证是通过一个通用的策略载体权限令牌（Rights Token）的推导生成的，权限令牌是一个独立于 DRM 的数据结构（P,C,U），它规定了主体 P 在使用模型 U 的限制下被允许访问内容资源 C，P 可以是一台设备或一组设备。CorallIF 定义了一系列针对权限令牌的操作框架，图 1-9 演示了使用权限令牌将一个 XML 格式的许可证推导出二进制许可证的过程。

CorallIF 主要由安全计算单元、安全消息协议和安全断言授权行为三部分组成。安全计算单元通常称为节点，每个节点关联一对公钥/私钥对，私钥认为是安全地存在于节点上。节点间通信采用 NEMO 安全消息协议，NEMO 协议框架使用标准的 Web 服务技术为通信节点提供点到点的消息机密性和完整性服务。安全断言授权行为在 CorallIF 中称为角色（Role），Role 本质上是一个签名的断言（Signed Assertion），指定节点通过它来证明和角色相关的某种行为。

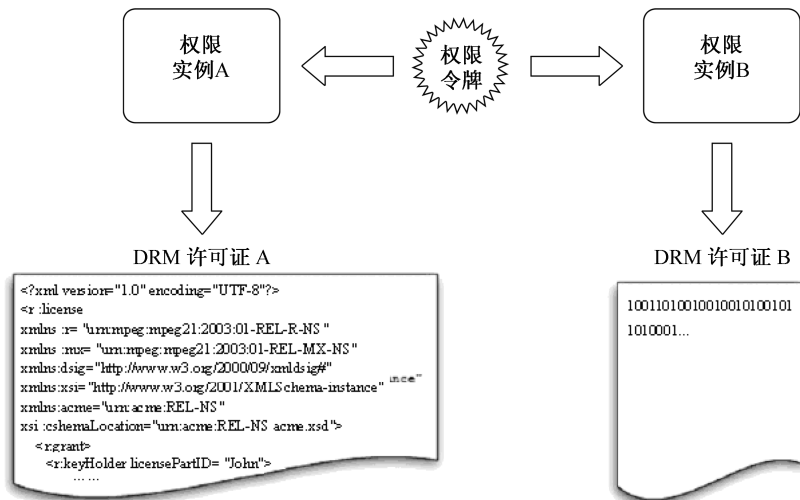


图 1-9 Coral DRM互操作实例

CoralIIF 的主要功能如下。

1. 获取权限令牌

用户访问喜爱的在线内容商店并购买内容。这样将会创建一个权限令牌 (P,C,U)，主体 P 指定由用户注册的一个特别设备集，使用模型 U 指定可以使用内容一段时间的权利。

2. 实例化

用户选择一个设备并请求将该权限令牌实例化。而互操作框架将执行下面的步骤：DRM 验证、确定主体、确定内容、确定权限令牌、创建许可证。在这个过程之后，这个不依赖于具体 DRM 的权限令牌将被转换为一个特定 DRM 证书。而所有这些步骤对于终端用户来说是透明的，用户仅仅知道他在购买内容几分钟后就能够在 he 选择的设备上来使用该内容了。

CoralIIF 中有三类角色。其中处理权利标志的角色负责在不同系统之间协调权限令牌的传输，处理主体的角色在名字空间内或名字空间之间管理主体之间的关系，而处理内容资源的角色主要为内容寻求合适的权利。此外，它还包含一些支持角色。

然而在 CoralIIF 中，Coral-enabled 服务器会记录所有用户的所有交易信息，存储在包含 Rights Registry 角色的节点中，此外该服务器还需要在 Principal Manager 中记录用户网络中所有设备信息。这些存储需求一方面增加了 Coral-enabled 服务器的负担，另一方面还有可能泄露用户隐私信息。

CoralIIF 提供了一套较为完善的工作机制，但该方案过于复杂且尚处于研究阶段，因而对于一般系统来说并非是最好的选择。

1.6.3 面向服务的框架

Filho 等研究者提出了一个面向服务的框架^[36]，该框架可提高系统之间的互操作性。框架中将系统涉及的权利和条件集中到一个单一的基于策略的模型内，该模型对每个平台都是通用的。为达到这个效果，它定义了一个面向服务的体系结构，该结构负责管理这些策略并用

它们来生成不同平台格式中的数字许可证。

如图 1-10 所示，该策略基于一个面向对象的模型，该模型可从概念上分为抽象的层次。较高的层次的基础是基于角色的访问控制（RBAC）及其扩展（GRBAC）。框架中不采用将权利和每个用户相联系的方式，而是将权利赋予主体角色，该角色可依次与用户相连接。这种方式使得较小的策略集就足以管理巨大而复杂的系统。

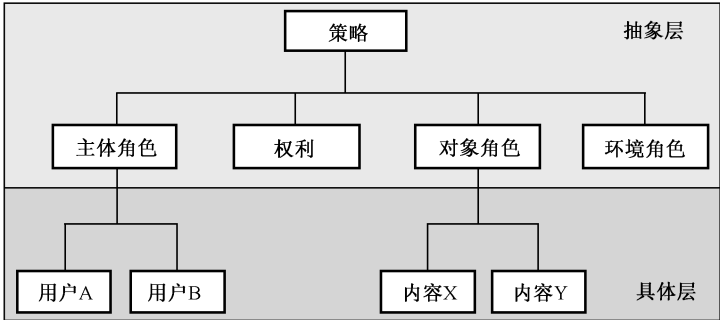


图 1-10 面向服务框架的策略结构

实际上，DRM 许可通常基于状态信息来和某个权利（如播放或打印）的条件和限制相关联。该信息被包含在数字许可证中并被一个特定的平台用来控制权利使用。GRBAC 通过引入环境角色来扩展 RBAC，该角色被应用到策略模型中以合并那些基于状态的条件和限制。GRBAC 还定义了对象角色，该角色被用来组织内容及建立基于内容特点的策略。第二个层次包含来自系统的具体实体（如用户、内容），并具有比上述层次更为动态的行为。

该框架是一个服务驱动的体系结构，由五种服务构成。其中一部分服务依赖于平台并通过框架的操作和系统接口，另一部分则和策略数据库进行互操作。

可以看到，该框架对系统具有通用性，但其对实现细节的描述不足，且对 DRM 系统中关键的安全性问题考虑不足。

1.6.4 DRM 内容改写体制

Nam 等研究者提出了一种方案^[37]，该方案能在 PAV（移动音视频）设备环境中将一 DRM 系统中的内容改写为另一个系统中的内容。

为了在两个域之间改写内容，必须改写格式和过程。该方案使用中间格式来完成这一点。中间格式的内容只在改写过程期间存在。必须保证改写过程在一个可信的环境中进行。下面对该方案的主要功能进行介绍。

1. DRM 内容改写过程

在源客户端，将 DRM 内容解包成资源、元数据和权利说明，并将其发送给源改写模块；源改写模块将收到的资源、元数据和权利说明打包成中间格式的中间 DRM 内容，并将其发送到目的改写模块；目的改写模块将收到的中间 DRM 内容，解包为资源、元数据和权利说明，并将它们发送到目的 DRM 的 PAV 设备管理器，该管理器将收到的资源、元数据和权利说明打包成本地内容，并将其发送到相应的设备。

2. 中间 DRM 内容的传输

中间格式的资源、元数据和权利说明将和一个包含附加信息的额外首部一起保存,这称为混合分发。混合分发中,有一个首部和一个包含资源、元数据和权利说明的主体,该首部中规定了用于单独处理中间资源、中间元数据和中间权利说明的位置,散列值基于首部和主体,但不包括散列值自身和数字签名。

3. 内容解密密钥的传输

该方案采用了 PKI (Public Key Infrastructure, 公钥基础设施) 机制来保证源改写模块和目的改写模块之间的安全内容传输,利用该机制,可保证密钥传输的安全性和完整性。内容解密密钥将会用对方的公钥进行加密,然后将它作为中间内容首部的一部分进行传输。

可以看到,该系统通过内容改写实现了系统之间的互相访问。但该系统中将改写模块放在用户端实现,这给用户端带来了较大的负担且存在一定的安全隐患。

参考文献

- [1] 俞银燕. 数字版权保护技术研究综述. 计算机学报. 2005, 28(12):1957-1968.
- [2] W. Rosenblatt, W. Trippe, S. Mooney Digital Rights Management: Business and Technology. New York: M&T Books, 2002.
- [3] 赵蕴华, 姚长青, 冯尉. 基于 DOI 的数字版权管理系统的模型研究. 情报理论与实践. 2010, 33(2): 102-105.
- [4] J. Dalziel. DOI in a DRM environment, A white paper for Copyright Agency Limited (CAL), DOI EPICS project, 2003.
- [5] P. Jamkhedkar, G. Heileman. DRM as a Layered System[C]. Proc of 2004 ACM Workshop on Digital Rights Management. New York: ACM Press, 2004:11-21.
- [6] 张志勇, 牛丹梅. 数字版权管理中数字权利使用控制研究进展. 计算机科学. 2011, 38(4):48-54.
- [7] J. Park, R. Sandhu. The UCONABC Usage Control Model. ACM Transactions on Information and System Security, 2004(1):128-174.
- [8] A. Pretschner, M. Hilty, F. Schutz, et al. Usage Control Enforcement: Present and Future. IEEE Security & Privacy, 2008, 6(4):44-53.
- [9] E. Diehl. A Four-layer Model for Security of Digital Rights Management. Proc. of 2008 ACM Workshop on DRM. New York.: ACM Press, 2008:19-27.
- [10] S. K. Nair, A. S. Tanenbaum, G. Gheorghe, et al. Enforcing DRM Policies Across Applications. Proc. of 2008 ACM Workshop on DRM. New York.: ACM Press, 2008:87-94.
- [11] B. C. Popescu, B. Crispo, F. Kamperman, et al. A DRM Security Architecture for Home Networks. Proc. of 4th ACM Workshop on Digital Rights Management. New York: ACM Press. 2004: 1-10.

- [12] H. Kim, Y. Lee, B. Chung, et al. Digital Rights Management with Right Delegation for Home Networks. LNCS 4296. Heidelberg: Springer Verlag, 2006: 233-245.
- [13] M. Barhoush, J. W. Atwood. Requirements for enforcing digital rights management in multicast content distribution. Telecommunication Systems, 2009.
- [14] T. Sans, F. Cuppens, C. B. Nora. OPA: Onion Policy Administration Model - Another Approach to Manage Rights in DRM. Proc. of 2007 IFIP International Federation for Information Processing. Heidelberg: Springer Verlag, 2007:349-360.
- [15] S. Bhatt, R. Sion, B. Carbunar. A Personal Mobile DRM Manager for Smartphones. Computers & Security, 2009, 28(6): 327-340.
- [16] S. Lee, J. Kim, S. J. Hong. Redistributing Time-based Rights Between Consumer Devices for Content Sharing in DRM System. International Journal of Information Security, 2009, 8(4): 263-273.
- [17] X. Feng, Z. Tang, Y. Y. Yu. An Efficient Contents Sharing Method for DRM. Proc. of 2009 Consumer Communications and Networking Conference. Washington DC: IEEE Press, 2009:1-5.
- [18] S. H. Kwok, S. M. Lui. A license management model to support B2C and C2C music sharing. In: Proceedings of the 10th International World Wide Web Conference Hong Kong, 2001, 136-137.
- [19] 邱罡, 王玉磊, 周利华. 基于可信计算的 DRM 互操作研究. 计算机科学. Vol.36 No.1, 2009, 1: 77-80.
- [20] E. Gallery, C. J. Mitchell. Trusted Mobile Platforms. LNCS 4677. Heidelberg: Springer Verlag, 2007: 282-323.
- [21] E. Gallery. Authorisation Issues for Mobile Code in Mobile Systems. London: Royal Holloway, University of London, 2007.
- [22] Y. Zheng. A Conceptual Architecture of a Trusted Mobile Environment. Proc. of the Second International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing. Washington DC: IEEE Press, 2006:75-81.
- [23] J. F. Reid, W. J. Caelli. DRM, Trusted Computing and Operating System Architecture. Proc. of 2005 Australasian Information Security Workshop. Washington DC: IEEE Press, 2005:127-136.
- [24] S. Stamm, N. P. Sheppard, N. R. Safavi. Implementing Trusted Terminals with a TPM and SITDRM. Electronic Notes in Theoretical Computer Science, 2008, 197(1): 73-85.
- [25] Z. Y. Zhang, Q. Q. Pei, J. F. Ma, et al. Implementing Trustworthy Dissemination of Digital Contents by Using a Third Party Attestation Proxy-Enabling Remote Attestation Model. Proc. of 2008 International Conference on MultiMedia and Information Technology. Washington DC: IEEE Computer Society Press, 2008:322-325.
- [26] TCG PC Client Specific TPM Interface Specification (TIS) Version 1.2
- [27] 张焕国等. 可信计算平台技术研究. 第十届全国容错计算学术会议论文集, 2003: 334-341.

- [28] 魏元首. 基于可信计算的数字版权管理. 西安电子科技大学硕士学位论文, 2008.
- [29] X. F. Chen, T. J. Huang. Interoperability issues in DRM and DMP solutions in: 2007 International Conference on Multimedia & Expo, 2007:907-910.
- [30] G. L. Heileman, P. A. Jamkhedkar. DRM Interoperability analysis from the perspective of a layered framework. DRM' 05-Proceedings of Fifth ACM Workshop on Digital Rights Management, 2005 : 17-26.
- [31] 李平. 数字版权管理系统及其协议研究. 华中科技大学博士学位论文, 2009.
- [32] A. Shamir. How to Share a Secret. Communications of ACM. 1979, 24(11): 612-613.
- [33] G. Gu, B. B. Zhu, S. Zhang. PLI: A New Framework to Protect Digital Content for P2P Networks. ACNS, pp.206-216, 2003.
- [34] S. O. Hwang, K. S. Yoon. Interoperable DRM framework for multiple devices environment. ETRI Journal, 2008, 30(4): 565-575.
- [35] T. Kalker, K. Carey, J. Lacy, et al. The coral DRM interoperability framework. In : 2007 4th Annual IEEE Consumer communications and Networking Conference, 2007: 930-934.
- [36] F. M. Figueira Filho, J. P. de Albuquerque, P. L. de Geus. A service-oriented framework to promote interoperability among DRM systems. In: Autonomic Management of Mobile Multimedia Services. 9th IFIP/IEEE International Conference on Management, 2006: 124-127.
- [37] D. W. Nam, Y. Jeong, J. Park, et al. DRM content adaptation between different DRM systems for seamless content service. In: Proceedings of the International Symposium on Consumer Electronics, ISCE, 2007: 867-870.

加密技术与数字签名技术

DRM 技术强调对数字内容的权限管理、转移以及二次分发等进行控制，但是，其对数字产品保护最基本、最低层的手段依然是加密技术。以密码学为基础的数据加密技术、数字签名技术、消息认证与身份识别技术是数字版权保护的核心技术，主要侧重于对数字内容的访问控制，解决数字内容的安全性问题，即通过对数字内容的加密，将数字内容的分发锁定和限制在特定用户的范围内，防止非授权访问。

加密技术主要分为对称密码体制和公钥密码体制。1949 年，Shannon 发表了“Communication Theory of Secrecy System”^[1]，对他所创立的信息论的概念和方法做了进一步发挥，并精辟地阐明了关于密码系统的分析、评价和设计的科学思想，是现代密码学的基础。1977 年 7 月 15 日，美国国家标准局（National Bureau of Standards, NBS）正式颁布数据加密标准（Data Encryption Standard, DES）^[2]。1976 年 Diffie 和 Hellman 在“New Directions in Cryptography”^[3]中第一次提出了公钥密码学概念，开创了密码学的新领域。1977 年麻省理工学院（Massachusetts Institute of Technology, MIT）MIT 三位年轻的科学家 Ron Rivest、Adi Shamir 和 Len Adleman 设计的 RSA 算法^[4]，是目前最有影响力的公钥密码算法。

数字签名通常用来进行身份、数据完整性、不可否认性的认证。身份认证是 DRM 系统的一个重要组成部分，是实施权限管理的基础。数字签名技术发展到今天，其理论研究和应用开发工作都得到了长足的发展。国际公认的数字签名算法主要有 RSA 算法^[4]、ElGamal 算法^[5]、ECC（Elliptic Curves Cryptography, 椭圆曲线密码）算法^[6]等。1994 年 5 月 19 日美国国家标准技术研究所（National Institute of Standards and Technology, NIST）公布了数字签名标准（Digital Signature Standard, DSS）^[7]，其中的数字签名算法 DSA（Digital Signature Algorithm）是 ElGamal 算法和 Schnorr 算法^[8]的变形。随着对数字签名研究和应用的深入，许多有特殊用途的数字签名方案被相继提出，如盲签名^[9]、群签名^[10]等。

当前部署公钥密码系统的主要手段是 PKI，PKI 是一种遵循既定标准的密钥管理平台，它能够为所有的网络应用透明地提供采用加密和数字签名等密码服务所必需的密钥和证书管理。利用 PKI 技术可以方便地建立和维护一个可信的网络计算环境，保证网上数据的机密性、完整性和有效性，确保电子交易有效、安全地进行。

文献[12]概述了目前 DRM 加密技术研究的三个主要方向，即 DRM 加密结构、加密算法和检验加密效果的方法，我们也将在本章做简单介绍。

2.1 密码学概述

2.1.1 密码体制与密码系统的基本模型

一个密码体制通常由五部分组成^[13]。

① 明文空间 M ：全体明文的集合。

② 密文空间 C ：全体密文的集合。

③ 密钥空间 K ：全体密钥的集合。通常每个密钥 $k \in K$ 都由加密密钥和解密密钥组成，即 $k = (k_e, k_d)$ ， k_e 与 k_d 可能相同，也可能不同。

④ 加密算法集合 E ：由加密密钥 k_e 控制的加密变换的集合。

⑤ 解密算法集合 D ：由解密密钥 k_d 控制的解密变换的集合。

设 $m \in M$ 是一个明文， $k = (k_e, k_d) \in K$ 是一个密钥，则有

加密过程： $c = E_{k_e}(m) \in C$ ；

解密过程： $m = D_{k_d}(c) \in M$ 。

其中， E_{k_e} 是由加密密钥 k_e 确定的加密变换， D_{k_d} 是由解密密钥 k_d 确定的解密变换。在一个密码体制中，为了使人们能够正常地进行加解密变换，必须要求解密变换是加密变换的逆变换，即 $\forall m \in M$ ，均有 $D_{k_d}(E_{k_e}(m)) = m$ 。

一个密码系统的基本模型如图 2-1 所示。

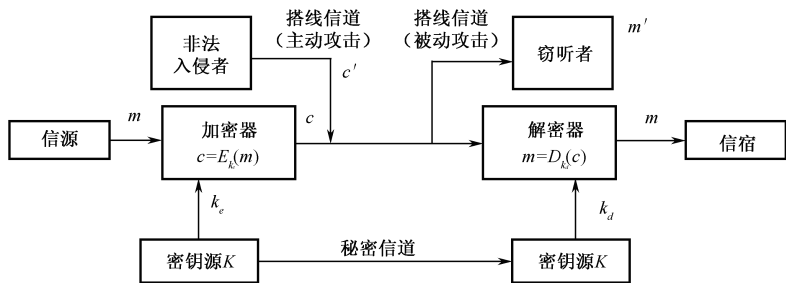


图 2-1 密码系统的基本模型

2.1.2 Kerckhoff 假设和密码系统的安全性

Shannon 在文献[1]中提出了无条件安全性（又称完善保密性）的概念：如果对于所有明文 P 和密文 C ，都有 $P_r(P) = P_r(P|C)$ 成立，则称该分组密码关于当前密钥具有无条件安全性，其中 $P_r(P)$ 表示明文 P 在消息空间上的分布概率， $P_r(P|C)$ 表示条件概率。在无条件安全性的模

型下，假定攻击者具有无限计算资源（如时间、空间、设备和资金等）。要达到无条件安全性，一个最基本的要求是：密钥长度至少要和待加密的消息的总长度相等，一个密钥比特只使用一次，这在绝大多数情况下是不切实际的。实际中，一个固定的密钥可以用来加密大量的明文块。

现代密码学中，通常是在计算安全性的模型下研究密码的安全性。一个密码系统是计算上安全的，指的是破译该系统所需的努力超越了攻击者的破译能力，或破译该系统的难度等价于求解数学上的某个已知难题。在计算安全性的模型下，假定攻击者拥有的计算资源是有限的。

Kerckhoff 假设：除了密钥之外，假定攻击者知道所有有关明文的统计特性、加密和解密的详细过程、密钥空间及其统计特性。**Kerckhoff 假设**意味着密码系统的安全性完全依赖于密钥的安全性。

在 **Kerckhoff** 假设下，根据攻击者所掌握的信息，密码分析者的攻击模型可分为以下几种^[13]。

- ① 唯密文攻击（Ciphertext-only Attack）：密码分析者仅知道一些密文。
- ② 已知明文攻击（Known-plaintext Attack）：密码分析者知道一些明文和相应的密文。
- ③ 选择明文攻击（Chosen-plaintext Attack）：密码分析者可以选择一些明文，并得到相应的密文。
- ④ 选择密文攻击（Chosen-ciphertext Attack）：密码分析者可以选择一些密文，并得到相应的明文。

其中唯密文攻击的攻击强度最弱，其他情况下的攻击强度依次增加。

2.1.3 分组密码的分析方法

一个攻击的有效程度通常由实施该攻击所需的时间复杂度、空间复杂度和数据复杂度来衡量。数据复杂度是实施该攻击所需输入的数据量，例如，已知明文攻击（或选择明文攻击）的数据复杂度可以用攻击中所需要的已知（或选择）明文-密文对的数量来确定。

对分组密码的常见攻击方式可分为以下几种^{[13][14]}。

1. 强力攻击

密码分析者通过试遍所有的密钥来进行破译，可用于任何密码攻击，攻击的复杂度依赖于分组长度和密钥长度。设 K 是密钥长度，在唯密文攻击下，攻击者依次试用密钥空间中的 2^K 个密钥解密一个或多个截获的密文，直到获取一个或多个有意义的明文块。在已知（选择）明文攻击下，攻击者依次试用密钥空间中的密钥对一个已知明文加密，将加密结果与该明文对应的密文比较，直至二者相等，然后再用其他几个已知明文-密文对验证该密钥的正确性。穷尽密钥搜索攻击的复杂度平均为 2^{K-1} 次。

显然，可以通过增大密钥量来对抗穷尽密钥搜索攻击。

2. 统计攻击

密码分析者通过分析明文和密文的统计规律来破译密码，是分组密码攻击的主要方法。差分密码分析和线性密码分析是统计分析攻击方法中最重要、最基本的两种方法。差分

密码分析的基本思想是通过分析明文对的差值对密文对的差值的影响来恢复某些密钥比特，线性密码分析的基本思想是根据密码体制中明文、密文以及密钥之间的线性逼近式的统计特性恢复某些密钥比特。

高阶差分密码分析、截断差分密码分析、不可能差分密码分析是差分密码分析的推广和补充。高阶差分密码分析一般对非线性模块的代数次数比较低、迭代轮数比较少的密码有效；对于某些密码体制，寻找高概率的差分几乎是不可能的，但只需要知道几比特差值的特性也可以恢复某些密钥比特，这就是截断差分密码分析的基本思想；不可能差分密码分析的基本思想是利用概率为 0（或非常小）的差分特征，排除那些导致概率为 0（或非常小）的差分特征所对应的密钥。

对抗统计分析攻击的方法是设法使明文的统计特性与密文的统计特性不一样。

3. 数学攻击

密码分析者针对加密变换的数学基础进行研究，通过数学求解的方法来设法找到相应的解密变换。为对抗这种攻击，应该选用具有坚实的数学基础和足够复杂的加密算法。

2.2 对称密码体制

在对称密码体制中，加密密钥 k_e 与解密密钥 k_d 相等。按照加密方式的不同又可分为两大类——分组密码和流密码，我们主要讨论分组密码。基于 Shannon 理论的 Feistel 密码结构^[15]，是当前大多数重要对称分组密码的基本结构。数据加密标准 DES 中的算法是第一个也是最重要的现代对称加密算法^[2]。

2.2.1 分组密码的设计思想与 Feistel 密码结构

设计一个分组长度为 n 的分组密码，其本质就是构造 $GF(2^n)$ 上的一个受密钥 k 控制的置换。给定一个密钥 k ，就得到一个具体的 $GF(2^n)$ 上的置换，不同的密钥应该对应不同的置换。

1949 年，Shannon 在“Communication Theory of Secrecy System”一文中提出了代换-置换网络（Substitution-Permutation Networks, S-P 网络）的思想，S-P 网络是基于代换和置换这两个基本操作的。而且 Shannon 认为，为了抵抗对手对密码体制的统计分析，必须对明文做混淆（Confusion）和扩散（Diffusion）处理，有效地隐藏明文的统计特性。混淆和扩散是现代分组密码的设计基础。

所谓混淆就是将密文与密钥、密文与明文之间的统计关系变得尽可能复杂，使得窃密者即使获取了关于密文的一些统计特性，也无法推测出密钥或明文。使用复杂的非线性变换可以达到比较好的混淆效果。

所谓扩散就是让明文中的每一个比特影响密文中的许多比特，或者说让密文中的每一比特受到明文中许多比特的影响，使明文统计结构扩散消失到大批密文统计特性中。

对于分组密码，在早期的研究中，基本上是基于 Feistel 结构进行的。Feistel 建议使用乘积密码的概念来逼近简单代换密码。乘积密码是指依次使用两个或两个以上基本密码，所得

结果的密码强度将强于所有单个密码强度。特别地，Feistel 建议交替地使用代换和置换。实际上，这是 Shannon 提出的交替使用混淆和扩散乘积密码的实际应用^[16]。图 2-2 描述了 Feistel 提出的密码结构。

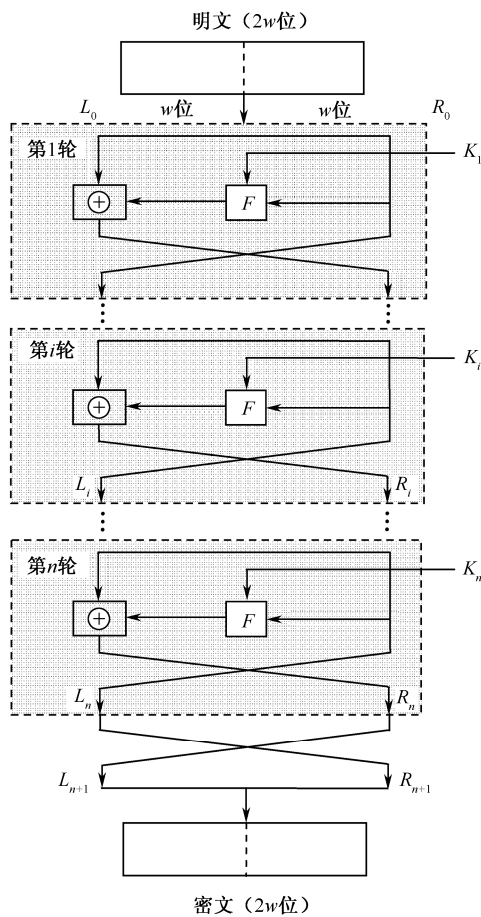


图 2-2 Feistel密码结构

Feistel 密码结构的输入是长为 $2w$ 的明文分组和密钥 K 。明文分组被分成为等长的两部分： L_0 和 R_0 。这两部分经过 n 轮迭代生成密文组。每 i 轮迭代的输入 L_{i-1} 和 R_{i-1} 来自于上轮迭代的输出；而输入的子密钥 K_i 是由整个密钥 K 推导出的。一般来说， K_i 不同于 K ，也互不相同。

每轮迭代具有相同的结构。代换作用在数据的左半部分。它通过用轮函数 F 对右半部分数据进行异或来完成。每轮迭代的轮函数是相同的，但是输入的子密钥 K_i 不同。代换之后，交换数据的左右两部分完成置换。这种结构是 Shannon 提出的 S-P 网络的一种特别形式^[16]。

Feistel 结构的具体实现依赖于以下参数和特征。

1. 分组长度

分组长度越长意味着安全性越高（其他数据不变），但是会降低加密和解密的速度。这种安全性的增加来自更好的扩散性。传统上，64 位的分组长度比较合理，在分组密码设计里很常用。然而，高级加密标准使用的是 128 位的分组长度。

2. 密钥长度

密钥长度较长同样意味着安全性较高，但会降低加密和解密的速度。这种安全性的增加来自更好的抗穷尽攻击能力和更好的混淆性。现在一般认为 64 位的密钥还不够，通常使用的密钥长度是 128 位。

3. 迭代轮数

Feistel 密码的本质在于单轮不能提供足够的安全性，而多轮加密可取得很高的安全性。迭代轮数的典型值是 16。

4. 子密钥产生算法

子密钥产生越复杂，密码分析攻击就越困难。

5. 轮函数

轮函数越复杂，抗攻击能力就越强。

Feistel 密码的解密算法与加密算法是相同的，只是子密钥的使用次序相反。因此不需要实现两个算法：一个做加密而另一个做解密。规则如下：将密文作为算法的输入，但是逆序使用子密钥 K_i 。也就是说，第一轮使用 K_n ，第二轮使用 K_{n-1} ，直到最后一轮使用 K_1 。

2.2.2 数据加密标准

1. DES 的结构

1973 年 5 月 13 日美国国家标准局 (National Bureau of Standards, NBS)，即现在的国家标准和技术协会 (National Institute of Standards and Technology, NIST) 在认识到建立数据保护标准既明显又急迫需要的情况下，开始征求国家密码标准方案。这一举措最终导致了数据加密标准 DES^[2] 的出现。NBS 提出密码算法标准包括：

- ① 算法必须是安全的；
- ② 算法必须是公开的；
- ③ 能够经济、有效地用硬件实现；
- ④ 能够得到批准；
- ⑤ 可出口。

在众多的算法中，IBM 公司 Tuchman Meyer 提出的算法 Lucifer 被选中。1975 年 3 月 17 日，NBS 公布了 IBM 公司提供的密码算法，以标准建议的形式在全国范围内征求意见。1977 年 7 月 15 日，NBS 宣布接受这个建议，即 DES (Data Encryption Standard) 正式颁布，DES 被 NBS 确定为联邦信息处理标准 (FIPS PUB 46)，供商业界和非国防性政府部门使用。几十年来，DES 得到了广泛的应用，尤其在金融领域发挥了重要的作用。

DES 采用了 64 位的分组长度和 56 位的密钥长度。除了初始置换和逆初始置换，DES 的结构 (图 2-3) 与图 2-2 所示的 Feistel 密码结构完全相同。

2. 初始置换和逆初始置换

DES 的初始置换表和逆初始置换表如表 2-1 所示。

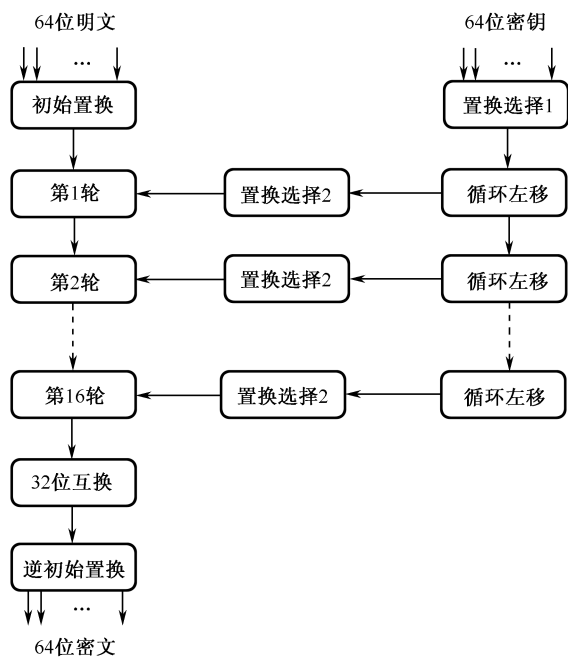


图 2-3 DES加密算法

表 2-1 DES 初始置换表和逆初始置换表

(a) 初始置换 (IP)							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

(b) 逆初始置换 (IP ⁻¹)							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

从表 2-1 可以看出，这两个置换是互逆的。对于 64 位的二进制分组数据 M ，经过置换 $X=IP(M)$ 后，再对它进行逆置换 IP^{-1} ，得到 $Y=IP^{-1}(IP(M))$ ，就会恢复出 M 。例如，经过 IP 置换后， M 的第 1 位被置换到第 40 位，再经过逆置换 IP^{-1} ，第 40 位又回到第 1 位的位置。

3. 迭代过程

图 2-4 给出了一轮迭代的内部结构，64 位的中间数据被分成了左右两部分，分别记为 L_i 和 R_i ($1 \leq i \leq 16$)。整个过程可以用下面的公式表示：

$$L_i=R_{i-1} \tag{2-1}$$

$$R_i=L_{i-1} \oplus F(R_{i-1}, K_i) \tag{2-2}$$

其中， K_i 为 48 位的子密钥。子密钥 K_1, K_2, \cdots, K_{16} 是作为密钥 K (56 位) 的函数而计算出的。

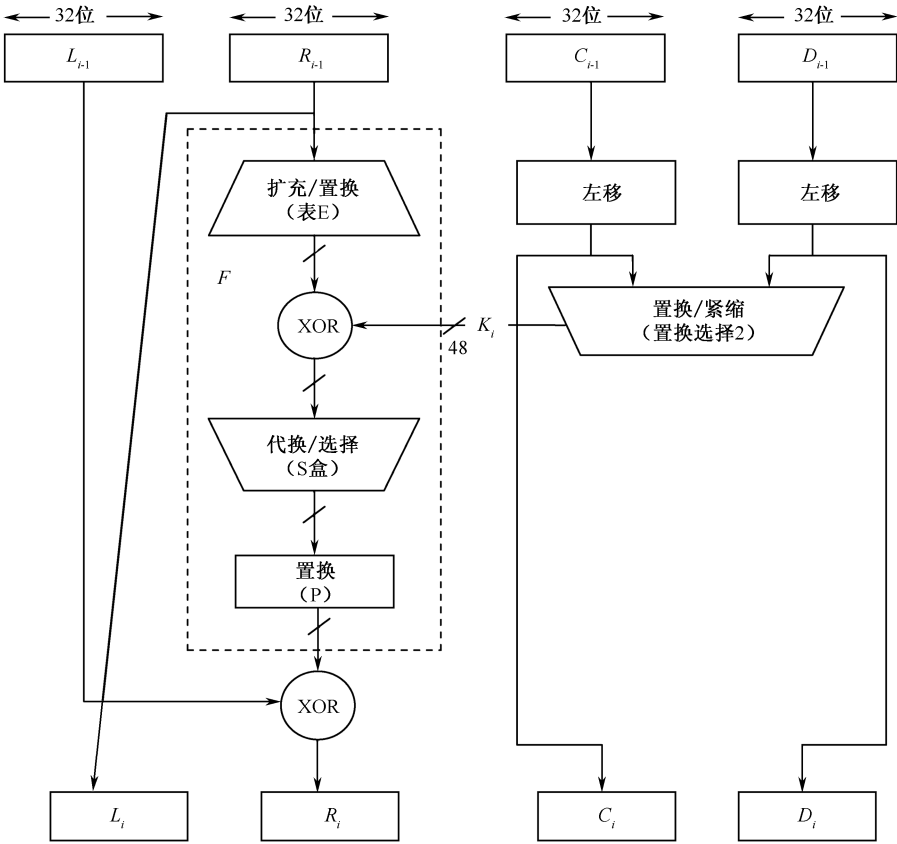


图 2-4 DES算法的一轮迭代过程

输入的 R_{i-1} 有 32 位，先被扩展到 48 位（表 2-2），扩展后所得到的 48 位结果再与 K_i 进行异或，这样得到的 48 位结果再经过一个变换函数（S 盒）产生 32 位输出，最后进行 P 置换（表 2-3）。

表 2-2 扩展置换表 (E)

31	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

表 2-3 直接置换表 (P)

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

4. S 盒的设计

S 盒代换是 DES 算法中最重要的部分，也是最关键的步骤，因为其他的运算都是线性的，易于分析，只有 S 盒代换是非线性的，它比 DES 中任何一步都提供了更好的安全性。

图 2-5 解释了 S 盒在函数 F 中的作用。代换函数由 8 个 S 盒组成，每个 S 盒输入 6 位，输出 4 位。

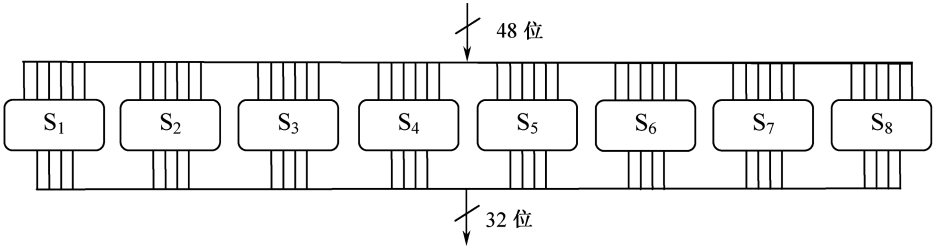


图 2-5 S 盒的结构

每个 S 盒代换参见表 2-4。其解释如下：S_i 盒输入的第一位和最后一位组成一个 2 位的二进制数用来选择 S 盒 4 行代换中的一行，中间 4 位用来选择 16 列中的某一列，行列交叉处的十进制数转换为二进制后可得到输出的 4 位二进制数。例如，在 S₁ 中，如输入为 101101，则行是 3(11)，列是 6(0110)，该处的值是 1，所以输出为 0001。

表 2-4 DES 的 S 盒代换表

S ₁	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S ₂	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S ₃	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S ₄	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S ₅	2	12	4	1	7	10	11	6	8	4	3	15	13	0	14	9
	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S ₆	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S ₇	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S ₈	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

S 盒的每行都定义了一个普通的可逆代换，图 2-6 显示了 S₁ 盒第 0 行的代换关系。

5. 密钥的产生

DES 子密钥的生成如图 2-3 和图 2-4 右半部所示。DES 算法输入了 64 比特的密钥（包括 8 位的奇偶校验位），密钥的各比特分别标记为 1 到 64，首先将每行的第 8 位，即奇偶校验位剔除 [表 2-5 (a)]，然后进行置换选择 1 [表 2-5 (b)]，将留下的 56 位密钥顺序按位打乱，所得 56 位密钥分为两个 28 位数据 C₀ 和 D₀。每轮迭代中，C_{i+1} 和 D_{i+1} 分别循环左移一位或两位 [表 2-5 (d)]，移位后的值作为下一轮的输入，它们同时也作为置换选择 2 [表 2-5 (c)] 的输入，将产生一个 48 位的输出作为函数 F(R_{i+1}, K_i) 的输入。

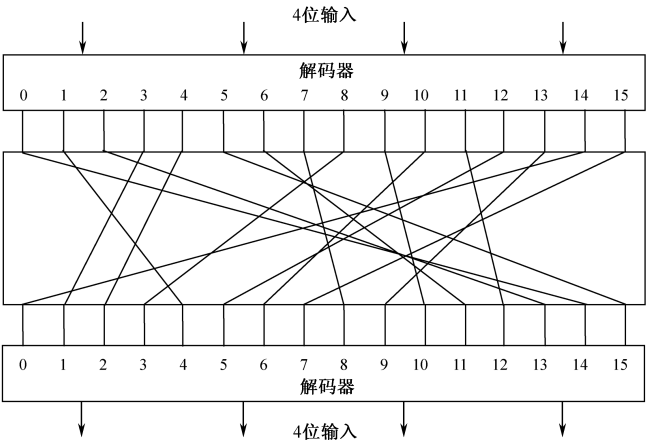


图 2-6 $n=4$ 时的一个 n 位到 n 位的分组密码

表 2-5 DES 子密钥的产生

(a) 输入密钥							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

(b) 置换选择 1						
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

(c) 置换选择 2					
14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

(d) 对左移次数的规定															
迭代次数	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
移位次数	1	1	2	2	2	2	2	2	1	2	2	2	2	2	1

6. DES 的安全性

这些年来,对 DES 的安全性一直争论不休,有许多人试图寻找并利用算法的弱点来进行攻击。对 DES 的争论焦点主要集中在密钥的长度和算法本身的安全性。

(1) 密钥长度

IBM 原来的 Lucifer 算法的密钥长度是 128 位,而提交作为标准的系统却只有 56 位,这个密钥长度不足以抵御穷举攻击。1977 年,Diffie 和 Hellman 提出制造一个每秒测试 10^6 的 VLSI 芯片,则一天就可以搜索完整个密钥空间,当时造价 2 千万美元。1997 年,美国科罗拉多州的程序员 Verser 借助互联网上 14000 多台计算机,花费 96 天用穷尽密钥搜索法破解了一个 DES 密钥。6 个月后,用这种方法破解 DES 的时间减少到 39 天。1998 年电子前沿基金会(Electronic Frontier Foundation, EFF)宣布一台造价不到 25 万美元的“DES 破译机”用三天时间破译了 DES,证明了 DES 是不安全的。

随着计算机速度的提高和硬件价格的下降,最终会导致 DES 毫无价值。幸运的是,有大量 DES 的代换算法,最重要的有高级加密标准(Advanced Encryption Standard, AES)和 3DES。

(2) DES 的半公开性

在 DES 里的所有计算中,除 S 盒之外全是线性的,作为唯一的非线性部件, S 盒对 DES 的安全性至关重要,是 DES 的核心。而 S 盒的设计标准,实际上包括整个算法的设计标准是不公开的,在 DES 刚提出时,就有人怀疑 S 盒里隐藏了“陷门”,使得美国国家安全局能够轻易地解密消息。后来表明 DES 里的 S 盒被设计成能够防止某些类型的攻击。在 20 世纪 90 年代初, Biham 与 Shamir 发现差分密码分析时,美国国家安全局承认某些未公布的 S 盒设计准则正是为了使得差分密码分析变得不可行。事实上,差分密码分析在 DES 最初研发时就被 IBM 的研究者所知,但这种方法却被保密了近 30 年,直到 Biham 与 Shamir 又独立地发现了这种攻击。

多年来人们的确发现了 S 盒的许多规律和一些缺点,尽管如此,还没有人发现 S 盒存在致命的弱点。

1998 年美国终于决定不再继续延用 DES 作为联邦加密标准, DES 将退出加密标准的舞台,寻找 DES 的替代者已刻不容缓,新的标准 AES 粉墨登场。

2.2.3 高级加密标准

1. AES 的评选过程

1999 年, NIST 发布了一个新版本的 DES 标准(FIPS PUB 46-3),该标准指出 DES 仅能用于遗留的系统,同时 3DES 将取代 DES 成为新的标准。然而,用软件实现该算法的速度比较慢;另外, DES 和 3DES 的分组长度均为 64 位,就效率 and 安全性而言,分组长度应更长。

由于这些缺陷,1997 年 1 月 2 日,美国的 NIST 开始了遴选 DES 替代算法的工作。该替代算法叫做高级加密标准,即 AES,要求安全性能不低于 3DES,同时具有更好的执行性能。除此之外, NIST 特别提出了高级加密标准必须是分组长度为 128 位的对称分组密码,并能支持长度为 128 位、192 位或 256 位的密钥,而且要求 AES 要能在全世界范围内免费得到或遵守与美国国家标准协会 ANSI 专利政策一致的规定获得。

令人始料不及的要求可能是 128 比特的分组长度。在此之前，几乎所有的分组密码设计都使用了与 DES 一样的 64 比特分组长度。此要求的提出是基于以下事实的考虑：其一，除了信息加密之外，分组密码还有消息认证等多种用途。在 n 比特分组的认证应用中，消息个数多于 $2^{n/2}$ 就容易出现认证冲突。其二，在分组密码的运行模式中， $2^{n/2}$ 经常作为一个安全界。如果攻击者得到多于 $2^{n/2}$ 个的密文，安全度就会下将。因此，对 AES 安全度增强的要求来说，加大分组长度至少和加大密钥长度一样重要^[17]。

AES 征集通知发出后，世界各国许多组织和机构反响强烈。全球 12 个不同国家的研究人员参与了竞争，开发了多种高级编码方法。到 1998 年 6 月 15 日算法提交期限之时，NIST 总共收到 21 个算法。NIST 得到了世界许多计算机安全公司及大学的密码专家的宝贵支持，对所有参选算法进行了评估。1998 年 8 月 20~22 日，NIST 在加拿大的 Ventura 召开了第一次 AES 候选会议，宣布接受了其中的 15 个算法为候选算法（表 2-6^[17]），公布了若干讨论结果。同时，邀请世界各国密码机构攻击这些候选算法并努力破译其编码，征求进一步的评价分析，以确定哪些算法可以进入第二轮评选。

表 2-6 AES 候选算法

国家	算法	提交者
澳大利亚	LOK197	Lawrie Brown, Josef Pieprzyk, Jennifer Seberry
比利时	Rijndael	Joan Daemen, Vincent Rijmen
加拿大	CAST-256	Entrust Technologies, Inc.
哥斯达黎加	DEAL	Richard Outerbridge, Lars Knudsen
法国	FROG	TecApro Internacional S.A.
德国	DFC	Centre National pour la Recherche Scienfifique CNRS
日本	MAGENTA	Deutsche Telekom AG
韩国	E2	Nippon Telegraph and Telephone Corporation (NTT)
美国	CRYPTON	Future Systems, Inc.
	HPC	Rich Schroepfel
	MARS	IBM
	RC6	RAS Laboratories
	SAFER+	Cylink Corporation
	Twofish	Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson
挪威	Serpent	Ros s Anderson, Eli Biham, Lars Knudsen

1999 年 3 月 22 日，NIST 在意大利罗马召开了第二次 AES 候选会议，提交了对 15 个候选算法的分析结果，讨论了它们的安全性、效率和设计灵活性等问题。第一轮评测于 1999 年 4 月 15 日结束，MARS、RC6、Rijndael、Serpent、Twofish 成为了终选算法。在随后开始的第二轮评测中，NIST 邀请密码专家对这 5 个终选算法进行强化攻击，会同世界密码界对各终选算法的安全性、速度及其通用性等要素进行评估。

2000 年 4 月 10~12 日，在纽约召开了第三次 AES 候选会议，提交并讨论 5 个终选算法的分析结果，公众评议时间截止于 2000 年 5 月 15 日。2000 年 10 月 2 日，美国商务部部长

Norman Y. Mineta 宣布推荐比利时密码学家 Joan daemen 博士和 Vincent Rijmen 博士提出的 Rijndael 算法为高级加密标准, 在全球范围内角逐了数年的激烈竞争结束。NIST 将就联邦信息处理标准草案进行 90 天的公众评论, 直至 2001 年 3 月 29 日。2001 年 11 月 26 日, NIST 完成评估并发布了最终标准 (FIPS PUB 197)。

2. Square 算法

NIST 和世界著名密码专家认为 5 个终选算法都具有很高的安全性, 之所以选择 Rijndael 算法, 主要因为它集安全、高效、性能、方便的使用及灵活性于一身。特别是 Rijndael 算法在不同硬件和软件运行环境下表现出始终如一的良好性能, 无论这些环境是否有反馈模式。它的密钥生成相当出色, 密钥的灵敏性也不错。Rijndael 算法对内存要求可以很低, 这使得它可以广泛使用于空间上受限制的环境, 并表现出出色的性能。Rijndael 成为了最能抵抗能量和时间攻击的算法之一, 在提供这些保护的同时没有对性能产生大的影响。从分组长度和密钥长度的观点来看, Rijndael 算法的设计带有灵活性, 同时还允许一定循环次数的修正。内部循环结构的设计使 Rijndael 算法能够在指令级上并行执行^[17]。

Rijndael 加密算法的原形是 Square 算法^[18], 它的设计策略是宽轨迹策略 (Wide Trail Strategy)。宽轨迹策略由 Joan Daemen 在其博士论文^[19]中提出, 是针对差分密码分析和线性密码分析制定的, 主要包括两个设计准则:

- ① 选择差分均匀性比较小和非线性度比较高的 S 盒;
- ② 适当选择线性变换, 使得固定轮数中的活动 S 盒的个数尽可能多。如果差分特征 (或线性逼近) 中某一轮的活动 S 盒的个数比较少, 那么下一轮中的活动 S 盒的个数就必须要多一些。

宽轨迹策略的最大优点是可以估算算法的最大差分特征概率和最大线性逼近概率, 由此可以评估算法抵抗差分密码分析和线性密码分析的能力^[14]。

Square 算法是 SPN 结构的分组密码。

- ① 初始密钥经过密钥编排算法得到轮密钥;
- ② 轮函数 σ 由列混淆 (M)、字节替换 (S)、矩阵转置 (T)、加密钥 (K) 四部分组成;
- ③ 明文和密文的长度都是 128 位, 并且它的密钥长度也是 128 位;
- ④ Square 算法的运算是在 4×4 的矩阵上进行的, 即不管是明文还是密文或是密钥, 都是把 128 位转换成 16 字节, 让这 16 字节按矩阵排列。

列混淆运算是指将一个已知矩阵的每一行与每一轮的每一列的输入相乘得到相应的列。四个新的列组成新的矩阵。这个已知的矩阵是:

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix}$$

这个运算是线性变换, 矩阵乘法是在有限域 $GF(2^8)$ 上进行的。例如:

$$\begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = \begin{pmatrix} 01 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix}$$

字节替换运算指经过乘矩阵运算后得到的矩阵的每个字节分别进入 S 盒,得到新的矩阵。公式为: $c_i = S(b_i)$ 。这里的 S 盒与 AES 中 S 盒相同。字节替换运算是数据置换的唯一非线性操作。

矩阵转置运算指将经过 S 盒运算得到的矩阵的列变行的操作。
加密运算指将矩阵转置后得到的矩阵与每一轮的密钥矩阵异或得到每一轮的输出,也为下一轮的输入。

Square 算法基本的结构图如图 2-7 所示。

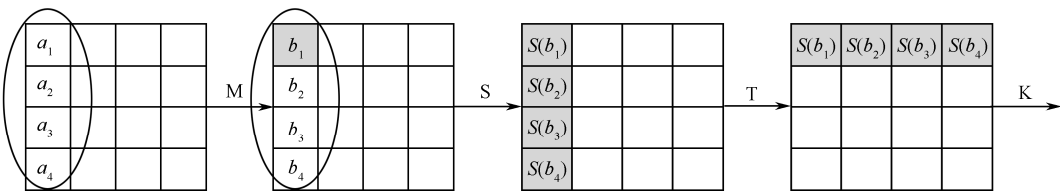


图 2-7 Square 算法的结构图

3. AES 结构

AES 分组密码的明文和密文的长度都是 128 位。AES 算法支持 128 位、192 位、256 位三种长度的密钥。AES 加密算法的明文、中间状态、密文及轮密钥都由二维的字节数组组成的 4×4 矩阵来表示,每一块为 8 比特,即 1 字节。密钥的长度不同,加密的轮数也不同,密钥长度 128 位、192 位、256 位相对应的轮数分别是 10 轮、12 轮和 14 轮。

图 2-8 是 AES 的完整结构^[16]。Rijndael 算法采用 SPN 结构,每一轮都使用代换和混淆并行地处理整个数据分组,因而扩散比 DES 算法更快。

128 位的消息分组被分成 16 字节,记为:

输入分组= m_0, m_1, \dots, m_{15}

密钥分组也是这样:

输入密钥= k_0, k_1, \dots, k_{15}

内部数据结构的表示是一个 4×4 矩阵:

输入分组= $\begin{pmatrix} m_0 & m_4 & m_8 & m_{12} \\ m_1 & m_5 & m_9 & m_{13} \\ m_2 & m_6 & m_{10} & m_{14} \\ m_3 & m_7 & m_{11} & m_{15} \end{pmatrix}$

输入密钥= $\begin{pmatrix} k_0 & k_4 & k_8 & k_{12} \\ k_1 & k_5 & k_9 & k_{13} \\ k_2 & k_6 & k_{10} & k_{14} \\ k_3 & k_7 & k_{11} & k_{15} \end{pmatrix}$

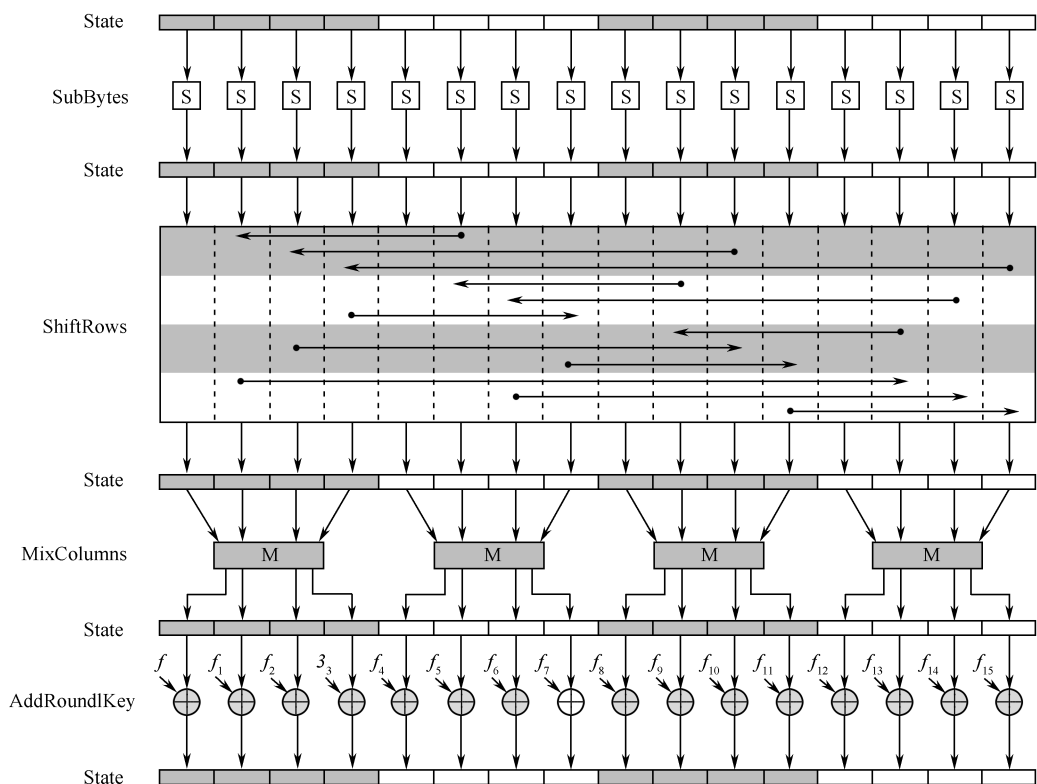


图 2-8 AES 的一轮加密过程

4. 迭代过程

同 DES 一样, Rijndael 算法也是由基本的变换单位——“轮”多次迭代而成的。在消息分组长度和密钥长度均为 128 比特的最小情况下, 轮数是 10。当消息长度和密钥长度变大的时候, 轮数也应该相应增加。

轮 (除了最后一轮) 变换由四个不同的变换组成:

```
Round(State, RoundKey){
    SubByte(State);
    ShiftRow(State);
    MixColumns(State);
    AddRoundKey(State, RoundKey);
}
```

这里 State 是轮消息矩阵, RoundKey 是轮密钥矩阵, 它是由输入密钥通过密钥表导出的。

字节替代 (SubBytes()) 的目的是得到一个非线性的代换密码。对于分组密码抗差分分析来说, 非线性是一个重要的性质。SubBytes() 变换是一个非线性的字节替代, 它独立地将 State 中的每个字节利用替代表 (S 盒) 进行运算。图 2-9 说明了 SubBytes() 变换在 State 上的作用效果。

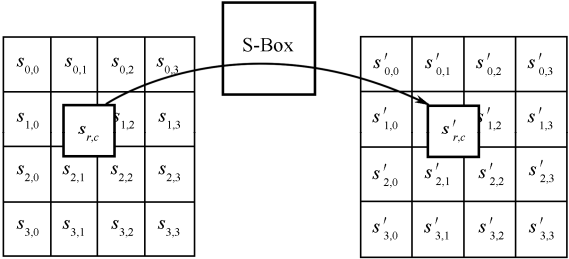


图 2-9 SubBytes()作用在State的单个字节上

行移位 (ShiftRows()) 和列混合 (MixColumns()) 的目的是获得明文消息分组在不同位置上的字节的混合, 消息分组不同位置上的字节的混合导致了消息在整个信息空间中更广的分布。

在 ShiftRows() 变换中, State 的最后 3 行循环移位不同的位移 r 。第一行中 $r = 0$, 即保持不变。

图 2-10 描述了 ShiftRows() 在 State 的后三行上的循环移位。

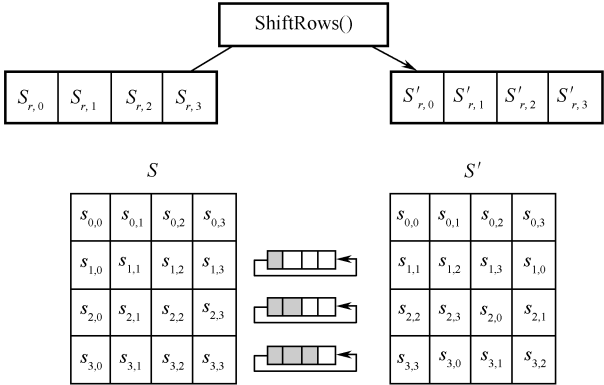


图 2-10 ShiftRows()在State的后三行上的循环移位

MixColumns() 变换在 State 上按照每一列进行运算, 图 2-11 描述了 MixColumns() 变换。

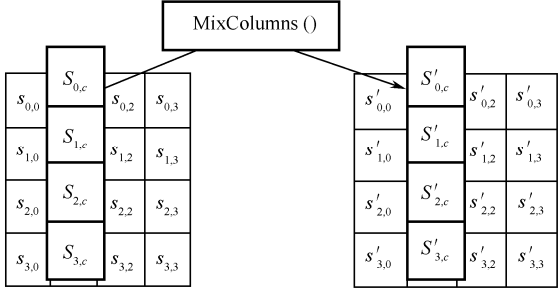


图 2-11 MixColumns()在State的列上运算

轮密钥加变换 (AddRoundKey()) 给出了消息分布所需的秘密随机性。在 AddRoundKey() 变换中, 用简单的比特异或将一个轮密钥作用在 State 上 (图 2-12)。

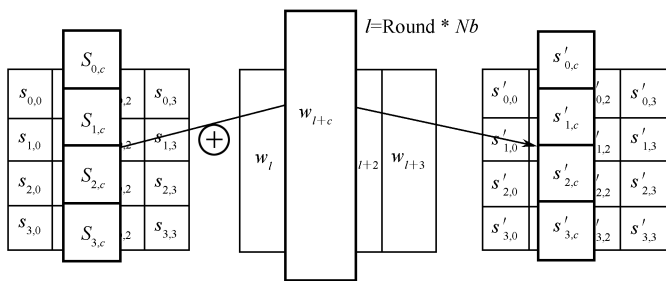


图 2-12 AddRoundKey() 将密钥编排得到的字异或到 State 的每一列上

四个内部函数都是可逆的，因此解密仅仅是在相反的方向反演加密，也就是说，运行：

```
AddRoundKey(State, RoundKey)-1;  
MixColumns(State)-1;  
ShiftRow(State)-1;  
SubByte(State)-1;
```

Feistel 密码的加密和解密可以使用同样的电路（硬件）和代码（软件），而 Rijndael 密码的加密和解密必须分别使用不同的电路和代码。

2.3 公钥密码体制

2.3.1 公钥密码的基本思想

在公钥密码体制以前的密码学发展史中，所有的密码算法，包括原始手工计算的、由机械设备实现的以及由计算机实现的，都基于代换和置换这两个基本方法。

公钥密码体制为密码学的发展提供了新的理论和技术基础，一方面公钥密码算法的基本工具不再是代换和置换，而是数学函数；另一方面公钥密码算法是以非对称的形式使用两个密钥，两个密钥的使用对保密性、密钥分配、认证等都有深刻的意义。可以说公钥密码体制的出现密码学史上是一次真正的革命。

公钥密码体制的概念是在解决单钥密码体制中最难解决的两个问题时提出的，这两个问题是密钥分配和数字签名。

1. 密钥分配问题

在传统的对称密码系统中，通信双方要进行保密通信，需要通过安全信道协商加密密钥，不知道这个密钥的第三方无法破解密文，以实现数据的保密性，这种安全信道可能很难实现。进入计算机网络时代后，对称密码体制便暴露出它的严重弱点，在有多用户的网络中，任何两个用户之间都需要有共享的密钥，若网上有 n 个用户，则需要密钥个数为 $c_n^2 = n(n-1)/2$ 。当网络中的用户数 n 很大时，需要管理的密钥数目是非常大的，如 $n=1000$ ， $c_n^2 = 499500$ 。因此密钥存储和管理都将是一个棘手的问题，而且安全性也得不到保证。

2. 数字签名问题

当主体 A 收到主体 B 的消息时，采用对称密码体制无法向第三方证明此消息确实来源于 B。

1976 年，Diffie 和 Hellman 首次提出了公钥密码的思想^[3]，他们开创性的工作引发了密码学的一场变革，标志着公钥密码学的诞生。Diffie 和 Hellman 提出的公钥密码算法的最大特点是采用两个相关密钥将加密和解密分开，其中一个密钥是公开的，称为公开密钥，简称公钥；另一个密钥为用户专用，是保密的，称为秘密密钥，简称私钥。因此公钥密码体制也称非对称密码体制。

公钥密码体制有以下重要特性：已知密码算法和公钥，求解密钥在计算上是不可行的。

1977 年，Rivest、Shamir 和 Adleman 三位 MIT 计算机科学实验室的研究人员设计了著名的 RSA 公钥密码系统^[4]；1978 年，Merkle 和 Hellman 基于背包问题实现了一个公钥加密系统^[20]；1985 年 T. ElGamal 提出了基于有限域上离散对数困难问题的 ElGamal 公钥密码^[5]；1986 年 Miller 提出了基于椭圆曲线上的离散对数困难问题的椭圆曲线密码体制^[6]。

2.3.2 背包加密算法

Shamir 于 1982 年用等价密钥产生超递增序列的方法^[21]，成功地破解了 Merkle 和 Hellman 的背包体制^[20]。虽然最初的背包加密法已经被破解了，但是，该方法所采用的 NP 完全问题的思想在非对称密码体制中被广泛地使用。

背包问题可以形象地描述。设一个体积为 t 的背包，以及一个元素集 $1, \dots, n$ ，其体积分别为 t_1, \dots, t_n ，是否存在这样一个子集，其体积之和等于背包的体积？例如，假设一个背包的体积为 20，元素集有 7 个元素，每个元素的体积分别为 5、3、7、1、12、15 和 19。哪些元素可以填满该背包？在这个示例中，有多个答案，第 4 个和第 7 个元素之和等于 20，第 1 个和第 6 个元素之和也等于 20。

☒ 定义 2-1 背包问题

给定一个数集 m_1, \dots, m_n ，一个目标总和 S ，是否能找出 b_1, \dots, b_n （其中 b_i 为 0 和 1），使得：

$$S = b_1 m_1 + b_2 m_2 + \dots + b_n m_n$$

对背包问题而言， S 即为总体积，如果 b_i 为 1，那么第 i 个元素就在背包中。

Merkle-Hellman 背包密码体制是以背包问题的形式来加密消息的。其背包以一系列的数字 m_1, \dots, m_n 给定，消息块是 n 个二进制位 b_1, \dots, b_n 的集合，密文就是前面给出的总和 S 。例如，给定一个背包列表 (5, 14, 9, 23, 16, 7, 31, 27)，字符 “a” 用 ASCII 码表示为 01100001，那么其密文就为：

$$S = (0 \times 5) + (1 \times 14) + (1 \times 19) + (0 \times 23) + (0 \times 16) + (0 \times 7) + (0 \times 31) + (1 \times 27) = 50$$

反过来，知道了密文，通过求解背包问题就可以还原明文。在上面的示例中，用穷尽搜索法还原明文也不太困难。由于其背包只有 8 个数字，对 b 而言，只有 $2^8 = 256$ 种 0 和 1 的

组合。但是,如果背包有 100 个元素,那么情况又会如何呢?这将有 2^{100} 种可能的组合。假设每秒检测一百万种组合,那么要找出一个正确的组合,平均要花费 5×10^{25} 秒的时间,大约等于 100 年。

背包问题所蕴含的思想是,利用寻找总和困难这一事实作为信息加密的一种方法。当然,合法用户必须能在正常的时间内解密信息,因此还必须具有解密算法,且只有合法用户才能使用。这种解密算法是基于一个简单的背包问题的。简单的背包具有一系列的数字,其中的每个数字都比其前面所有数字之和要大。(3, 5, 9, 18, 38, 75, 155, 312) 就是这样的一个数字系列。注意,9 比 3+5 大,18 比 3+5+9 大等。要加密字符“R”,首先将它用 ASCII 码表示,即为 01010010,因此目标总和为 5+8+155 (即 178)。对于这种形式的背包,从总和中可以很容易还原数列中的各个元素,从而很容易还原字符“R”。在上面的示例中,312 不可能参与总和的计算,因为目标总和比 312 要小。而 155 必须参与总和的计算,因为 155 之前所有数字之和只有 148,比 178 小。知道了 155 是目标总和 178 的一部分,就可以将它从总和中减去,即 $178-155=23$ 。现在,很显然,75 和 38 都不可能参与总和的计算,因为它们都比 23 大。因此,18 肯定是目标总和的组成部分。同样, $23-18=5$,这必定就是参与总和 178 的最后一个数了。从而可以知道,被加密字符的 ASCII 码为 01010010,即为字符“R”。

我们将上面这种简单的背包称为超递增背包。

☒ 定义 2-2 超递增背包

背包向量 $M=(m_1, \dots, m_n)$ 称为超递增的,如果

$$m_j > \sum_{i=1}^{j-1} a_i, \quad j=2, \dots, n \quad (2-3)$$

超递增背包向量对应的背包问题很容易通过贪婪算法求解。

👉 算法 2-1 贪婪算法

已知 S 为背包容量,对超递增背包 $M=(m_1, \dots, m_n)$ 从右向左检查每一元素,以确定是否在解中。若 $S \geq m_n$, 则 m_n 在解中,令 $b_n=1$; 若 $S < m_n$, 则 m_n 不在解中,令 $b_n=0$ 。下面令

$$S = \begin{cases} S, & S < m_n \\ S - m_n, & S \geq m_n \end{cases} \quad (2-4)$$

对 m_{n-1} 重复上述过程,一直下去,直到检查出 m_1 是否在解中。检查结束后得 $B=(b_1, \dots, b_n)$ 。

然而,敌手如果也知道超递增背包向量,同样也很容易解密。为此可用模乘对 M 进行伪装,模乘的模数 k 和乘数 t 皆取为常量,满足 $k > \sum_{i=1}^n m_i$, $\gcd(t, k)=1$, 即 t 在模 k 下有乘法逆元。

设

$$m'_i \equiv t \cdot m_i \bmod k, \quad i=1, 2, \dots, n \quad (2-5)$$

得一新的背包向量 $M'=(m'_1, \dots, m'_n)$, 记为 $M'=t \cdot M \bmod k$, 用户以 M' 作为自己的公开密钥。

两年之后,该加密算法被破解,破解的基本思想是不必找出正确的模数 k 和乘数 t (即陷门信息),只需找出任意模数 k' 和乘数 t' ,使得用 k' 和 t' 去乘公开的背包向量 M' 时,能够产生超递增的背包向量即可。

2.3.3 RSA 算法

RSA 算法是 1978 年由 Rivest、Shamir 和 Adleman 提出的一种用数论构造的公钥密码体制^[4],得到了广泛的应用。这种密码体制的安全性基于大整数的因数分解的困难性,是迄今为止最为成熟完善的算法。

RSA 公钥密码体制表述如下:设模数 $n=pq$ (p, q 是两个大素数),设明文空间为 P ,密文空间为 C ,满足 $P=C=Z_n$,定义密钥集:

$$K=\{(n, p, q, e, d): ed \equiv 1 \pmod{\varphi(n)}\} \quad (2-6)$$

其中, $\varphi(n)$ 为欧拉函数, e 是加密指数,满足 $1 < e < \varphi(n)$,且 $\gcd(\varphi(n), e) = 1$; d 是解密指数。对于 $k=(n, p, q, e, d) \in K$,定义对明文 m 的加密算法和对密文 c 的解密算法为:

$$c \equiv m^e \pmod{n} \quad (2-7)$$

$$m \equiv c^d \pmod{n} \quad (2-8)$$

其中, $x, y \in Z_n^*$, (n, e) 组成公钥, (p, q, d) 为私钥。

算法 2-2 RSA 密钥产生算法

- ① 选取两个秘密的大素数 p 和 q ;
- ② 计算 $n=p \times q, \varphi(n)=(p-1)(q-1)$, 其中 $\varphi(n)$ 是 n 的欧拉函数;
- ③ 选一随机整数 e , 满足 $1 < e < \varphi(n)$, 且 $\gcd(\varphi(n), e) = 1$;
- ④ 计算 d , 满足 $d \cdot e \equiv 1 \pmod{\varphi(n)}$, 即 d 是 e 在模 $\varphi(n)$ 下的乘法逆元;
- ⑤ 以 (n, e) 为公钥, (p, q, d) 为私钥。

欧拉函数 $\varphi(n)$ 是指小于 n 且与 n 互素的正整数的个数,例如 $\varphi(1)=1, \varphi(35)=24$ 。我们列出所有小于 35 且与 35 互素的正整数如下: 1, 2, 3, 4, 6, 8, 9, 11, 12, 13, 16, 17, 18, 19, 22, 23, 24, 26, 27, 29, 31, 32, 33, 34。由上可知共有 24 个数,因此 $\varphi(35)=24$ 。

虽然 $\varphi(1)$ 的值无意义,但我们仍将它定义为 1。

显然,对素数 p 有:

$$\varphi(p)=p-1$$

假设有两个素数 p 和 $q, p \neq q$, 那么对 $n=pq$, 有

$$\varphi(n)=\varphi(pq)=\varphi(p) \times \varphi(q)=(p-1)(q-1)$$

算法 2-3 RSA 密码算法

加密时先将明文比特串分组,使得每个分组对应的十进制数小于 n ,即分组长度小于 $\log_2 n$,然后用式 (2-7) 对每个明文分组 m 做加密运算。

解密时用式 (2-8) 对每个密文分组 c 做解密运算。

在证明解密算法能还原出明文之前，我们先介绍欧拉定理。

★ 定理 2-1 欧拉定理

对任意互素的 a 和 n ，有

$$a^{\varphi(n)} \equiv 1 \pmod{n}$$

例如：

$$a=3, n=10, \varphi(10)=4$$

$$a^{\varphi(n)} = 3^4 = 81 \equiv 1 \pmod{10} = 1 \pmod{n}$$

$$a=2, n=11, \varphi(11)=10$$

$$a^{\varphi(n)} = 2^{10} = 1024 \equiv 1 \pmod{11} = 1 \pmod{n}$$

证明

由加密过程知 $c \equiv m^e \pmod{n}$ ，所以

$$c^d \pmod{n} \equiv n^{ed} \pmod{n} \equiv m^{k\varphi(n)+1} \pmod{n}$$

下面分两种情况：

① m 与 n 互素，则由欧拉定理得

$$m^{\varphi(n)} \equiv 1 \pmod{n}, m^{k\varphi(n)} \equiv 1 \pmod{n}, m^{k\varphi(n)+1} \equiv m \pmod{n}$$

即

$$c^d \pmod{n} \equiv m \quad (2-9)$$

② $\gcd(m, n) \neq 1$ ，先看 $\gcd(m, n)=1$ 的含义，由于 $n=pq$ ，所以 $\gcd(m, n)=1$ 意味着 m 不是 p 的倍数也不是 q 的倍数，因此 $\gcd(m, n) \neq 1$ 意味着 m 是 p 的倍数或 q 的倍数。不妨设 $m=tp$ ，其中 t 为一正整数。此时必有 $\gcd(m, q)=1$ ，否则 m 也是 q 的倍数，从而是 pq 的倍数，与 $m < n=pq$ 矛盾。

由 $\gcd(m, q)=1$ 及欧拉定理得 $m^{\varphi(q)} \equiv 1 \pmod{q}$ ，所以

$$m^{k\varphi(q)} \equiv 1 \pmod{q}, [m^{k\varphi(q)}]^{\varphi(p)} \equiv 1 \pmod{q}, m^{k\varphi(n)} \equiv 1 \pmod{q}$$

因此存在一整数 r ，使得 $m^{k\varphi(n)} = 1 + rq$ ，两边同乘以 $m=tp$ 得

$$m^{k\varphi(n)+1} = m + rtpq = m + trn$$

即

$$m^{k\varphi(n)+1} \equiv m \pmod{n}$$

所以 $c^d \pmod{n} \equiv m$ 。

如前所述，RSA 算法是靠大整数因数分解的困难来保证其安全性的。下面对 RSA 的安全性进行分析。

- ① 密码分析者攻击 RSA 体制的关键点在于如何分解 n ；
- ② 若分解成功使 $n=pq$ ，则可以算出 $\varphi(n)=(p-1)(q-1)$ ，然后由公开的 e ，解出秘密的 d ；
- ③ 若要使 RSA 安全， p 与 q 必为足够大的素数，使分析者没有办法在多项式时间内将 n 分解出来。

目前对 RSA 攻击的手段有以下几种。

1. 强力攻击

大数分解是一个 NP 问题，目前已知的最好的算法需要进行 e^x 次算术运算。假设我们用一台每秒运算一亿次的计算机来分解一个 200 位十进制整数，则需要 3.8×10^7 年；类似地，要分解一个 300 位的十进制整数，则需要 4.86×10^{13} 年。Pentium100 大约是 125MIPS，它分解 RSA-129 需要 37 年，100 台 Pentium100 需要 4 个月。可见，增加位数，将大大地提高体制的安全性。

直接分解一个大素数的强力攻击的一个实例是：1994 年 4 月分解的 RSA 密钥 RSA-129，即分解了一个 129 位十进制、425 位的大素数。分解时启用了 1600 台计算机，耗时 8 个月，处理了 4600MIPS 年的数据（1MIPS 年是 1MIPS 的计算机一年所能处理的数据量）。

2. 数学攻击

对 RSA 最明显的攻击方式就是对模 n 进行因式分解。假如知道了模 n 的两个因子 p 和 q ，那么我们可以通过计算 $d \cdot e \equiv 1 \pmod{\phi(n)}$ 来获得私钥 d ，此时这个 RSA 密码体制就彻底瓦解了。

关于 RSA 的因式分解算法有大量的文献，但在现实中大整数分解最有效的三种算法是二次筛法（Quadratic Sieve）、椭圆曲线分解算法（Elliptic Curve Factoring）和数域筛法（Number Field Sieve），这三种算法在文献[22]中都有详细的介绍。

3. 计时攻击

对 RSA 还有一种攻击方法是计时攻击，这是 Kocher 于 1996 年发现的，具体的攻击方法可参见文献[23]。

就目前的计算机水平用 1024 位的密钥是安全的，2048 位是绝对安全的。RSA 实验室认为，512 位的 n 已不够安全，应停止使用，现在的个人需要用 668 位的 n ，公司要用 1024 位的 n ，极其重要的场合应该用 2048 位的 n 。

由于 RSA 算法进行的都是大整数的模幂运算，涉及素性检测、乘法逆元、加密和解密，因此无论是软件还是硬件实现，速度一直是 RSA 算法的缺陷。硬件实现时，RSA 比 DES 要慢大约 1000 倍，软件实现时，RSA 比 DES 要慢大约 100 倍。可见，用 RSA 直接加密信息有诸多不便，所以，很多实际系统中，只用 RSA 来交换 DES 的密钥，而用 DES 来加密主体信息。

2.3.4 ElGamal 算法

ElGamal 算法^[5]是密码学家 T. ElGamal 于 1985 年提出的。它的基本思想是基于有限域上求解离散对数的困难性，它既能用于数据加密也能用于数字签名，在现代密码学中应用十分广泛，是目前研究的热点之一。

定义 2-3 离散对数问题

给定一个素数 p , 两个整数 a 和 b , 且对于某些整数 m 有 $b = a^m \bmod p$ 成立。该问题是要找出一个 m , 也就是说, 求解方程 $m = \log_a b$ 。

如果 p 较小, 该问题就可以用穷尽搜索法来解决。例如, 假设 $p=17$, $a=3$, $b=5$, 求解满足 $5=3^m \bmod p$ 的 x 值。从 1 开始, 尝试所有可能的值, 直到找到一个满足该等式的结果:

x	1	2	3	4	5
3^x	3	9	10	13	5

因此, $m=5$ 。只要 p 比较小, 用穷尽搜索法就可以很容易求解该问题。如果 p 比较大, 假设是一个 100 位的素数, 那么就像整数的因子分解问题那样很难求解了。

欧拉定理告诉我们, 对任何互素的 a 和 n , 有:

$$a^{\varphi(n)} \equiv 1 \bmod n \quad (2-10)$$

其中, 欧拉函数 $\varphi(n)$ 是指小于 n 且与 n 互素的正整数的个数。下面我们考虑欧拉定理更一般的表示形式:

$$a^m \equiv 1 \bmod n \quad (2-11)$$

若 a 与 n 互素, 则至少有一个正整数 m 满足式 (2-11), 即 $m=\varphi(n)$ 。

我们称使式 (2-11) 成立的最小正幂 m 为 a 模 n 的阶或 a 所产生的周期长。考虑下面这个例子:

$$\begin{aligned} 7^1 &\equiv 7 \bmod 19 \\ 7^2 &= 49 = 2 \times 19 + 11 \equiv 11 \bmod 19 \\ 7^3 &= 343 = 18 \times 19 + 1 \equiv 1 \bmod 19 \\ 7^4 &= 2401 = 126 \times 19 + 7 \equiv 7 \bmod 19 \\ 7^5 &= 16807 = 884 \times 19 + 11 \equiv 11 \bmod 19 \end{aligned}$$

由于 $7^3 \equiv 1 \bmod 19$, 可得 $7^{3+j} \equiv 7^3 7^j \equiv 7^j \bmod 19$, 这说明若 7 的两个指数相差 3 (或 3 的倍数), 则以它们为指数的 7 的模 19 幂是相同的。换句话说, 该序列是周期性的, 且其周期长是使 $7^m \equiv 1 \bmod 19$ 成立的最小正幂 $m=3$ 。

更一般地, 我们说 $\varphi(n)$ 是一个数所属的模 n 的可能的最高指数。如果一个数的阶为 $\varphi(n)$, 则称之为 n 的本原根。

本原根的重要之处在于, 若 a 是 n 的本原根, 则其幂

$$a, a^2, \dots, a^{\varphi(n)}$$

是模 n 各不相同的, 且均与 n 互素。特别地, 对素数 p , 若 a 是 p 的本原根, 则

$$a, a^2, \dots, a^{p-1}$$

是模 p 各不相同的。素数 19 的本原根为 2, 3, 10, 13, 14 和 15。

并不是所有的整数都有本原根。事实上, 只有形为 2, 4, p^a 和 $2p^a$ 的整数才有本原根, 这里 p 是素数, a 是正整数。

算法 2-3 ElGamal 密钥生成算法

令 z_p 是一个有 p 个元素的有限域, p 是一个素数, g 是 z_p^* (z_p 中除去 0 元素) 中的一个本原元。选定私钥为 $\alpha(\alpha < p)$ 。计算公钥

$$\beta \equiv g^\alpha \bmod p \tag{2-12}$$

设明文集 M 为 z_p^* , 密文集 C 为 $z_p^* \times z_p^*$ 。

算法 2-4 ElGamal 加密算法

选择随机数 $k \in z_{p-1}$, 且 $(k, p-1)=1$, 计算:

$$y_1 = g^k \bmod p \quad (\text{随机数 } k \text{ 被加密}) \tag{2-13}$$

$$y_2 = M\beta^k \bmod p \quad (\text{明文被随机数 } k \text{ 和公钥 } \beta \text{ 加密}) \tag{2-14}$$

式中, M 是发送明文组, 密文由上述两部分 y_1 、 y_2 级联构成, 即密文 $C=y_1||y_2$ 。

密文由明文和所选随机数 k 来确定, 因而是非确定性加密, 一般称之为随机化加密, 对同一明文由于不同时刻的随机数 k 不同而给出不同的密文。这样做的代价是使数据扩展一倍。

算法 2-5 ElGamal 解密算法

收到密文 C 后, 计算明文

$$M = \frac{y_2}{y_1^\alpha} = \frac{M\beta^k}{g^{ka}} \equiv \frac{Mg^{ak}}{g^{ka}} \bmod p \tag{2-15}$$

例如, 选 $p=2579$, $g=2$, $\alpha=765$, 计算出 $\beta=g^{765} \bmod 2579=949$ 。若明文组为 $M=1299$, 选随机数 $k=853$, 可计算出 $y_1 \equiv 2^{853} \bmod 2597=435$ 及 $y_2 \equiv 1299 \times 949^{853} \bmod 2597=2396$ 。密文 $C=(435, 2396)$ 。解密时由 C 可算出消息组 $M \equiv 2396/(435)^{765} \bmod 2579=1299$ 。

在 ElGamal 密码算法的实际应用中, 和大多数公钥算法一样, 如何实现一个有效的大整数幂模运算是实现密码算法的关键。所有使用者可以选取使用同样的素数 p 和生成元, 在这种情况下, p 不必作为公钥的一部分而发布, 这可使公钥的长度小一些。使用固定元素还有另外的优点: 可以通过预计算来加快取幂运算, 但使用共同系统参数的一个潜在的缺点是必须保证模 p 足够大。

2.3.5 椭圆曲线加密算法

椭圆曲线作为代数几何中的重要问题已有 100 多年的研究历史, 积累了大量的研究文献, 但直到 1985 年, N. Koblitz 和 V. Miller 才独立将其引入密码学中, 成为构造公钥密码体制的一个有力工具。它是利用有限域上的椭圆曲线有限群代替离散对数问题中的有限循环群后得到的一类密码体制。由于椭圆曲线密码具有安全性能高、处理速度快、带宽要求低和存储空间小等特点, 与 RSA 相比, ECC 在密钥长度和运算速度上具有优越性。

椭圆曲线可以用三次方程来表示:

$$y^2+axy+by=x^3+cx^2+dx+e \tag{2-16}$$

其中, a, b, c, d 和 e 是实数, x 和 y 在实数集上取值。对加密而言, 将方程限制为下面的形式就已经足够了:

$$y^2 = x^3 + ax + b \quad (2-17)$$

在椭圆曲线中定义一个称为无穷远点或零点的元素, 记做 O 。考虑满足式 (2-17) 的所有点 (x, y) 和元素 O 所组成的点集 $E(a, b)$, 在式 (2-17) 中, 参数 a 和 b 如果满足条件:

$$4a^3 + 27b^2 \neq 0 \quad (2-18)$$

则可基于集合 $E(a, b)$ 定义一个群。

为了在 $E(a, b)$ 上定义一个群, 我们定义一个加法运算, 用 $+$ 表示, 其中 a 和 b 满足式 (2-18)。用几何术语可定义加法的运算规则: 若椭圆曲线上的三个点同在一条直线上, 则它们的和为 O 。从这个定义出发, 可以定义椭圆曲线加法的运算规则:

① O 是加法的单位元。这样有 $O = -O$; 对椭圆曲线上的任何一点 P , 有 $P + O = P$ 。

下面假定 $P \neq Q$ 且 $Q \neq O$ 。

② 点 P 的负元是具有相同 x 坐标和相反的 y 坐标的点, 即若 $P = (x, y)$, 则 $-P = (x, -y)$ 。注意这两个点可用一条垂直的线连接起来, 并且 $P + (-P) = P - P = O$ (图 2-13)。

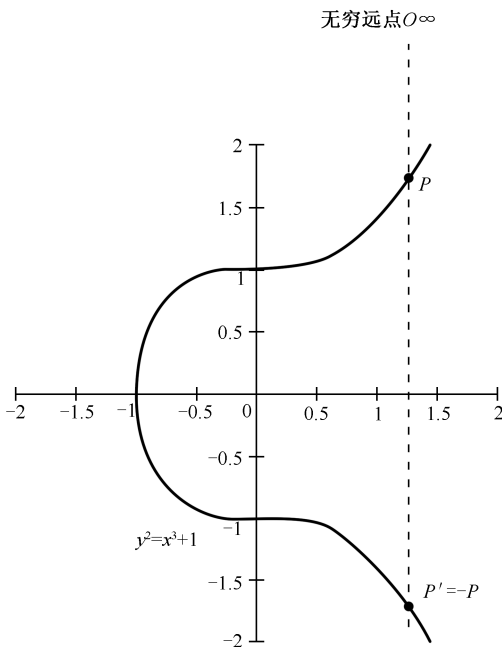


图 2-13 椭圆曲线上定义 $P + (-P) = O$

③ 要计算 x 坐标不相同的两点 P 和 Q 之和, 则在 P 和 Q 间画一条直线并找出第三个交点 R' , 显然存在有唯一的交点 R' (除非这条直线在 P 或 Q 处与该椭圆曲线相切, 此时我们分别取 $R' = P$ 或 $R' = Q$)。定义 $P + Q$ 为 R' (相对于 x 轴) 的镜像 R (图 2-14)。

④ 上述术语的几何解释也适用于具有相同 x 坐标的两个点 P 和 $-P$ 的情形 (图 2-15)。用垂直的线连接这两点, 这也可看做是在无穷远点处与曲线相交, 因此有 $P + (-P) = O$, 与上述定义一致。

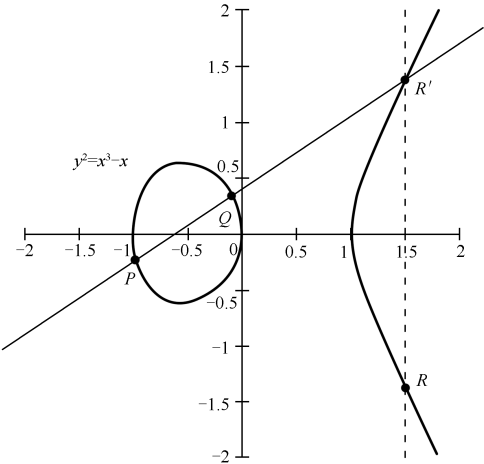


图 2-14 椭圆曲线上的加法 $P+Q=R$

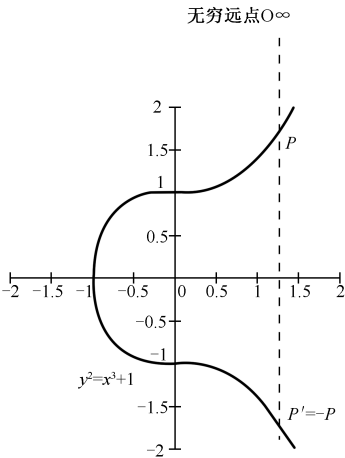


图 2-15 椭圆曲线上的加法 $P+(-P)=O$

⑤ 为计算点 Q 的两倍，画一条切线并找出另一交点 R' ， R 为 R' （相对于 x 轴）的镜像，则 $Q+Q=2Q=R$ （图 2-16）。

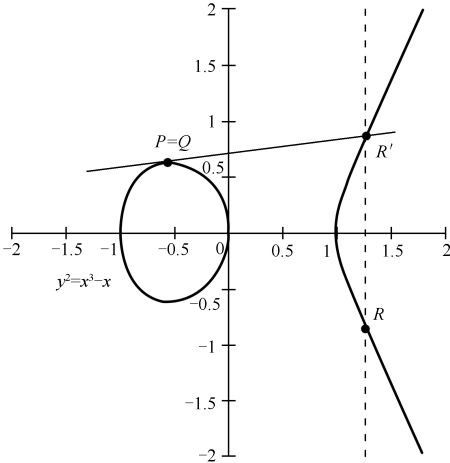


图 2-16 椭圆曲线上的加法 $Q+Q=R$

可以证明，集合 $E(a, b)$ 在上述运算规则下，构成阿贝尔群。

要建立基于椭圆曲线的密码体制，需要类似因子分解两个素数之积或求离散对数这样的“难题”。对于椭圆曲线，这种问题的描述如下：给定两个点 P 和 W ，其中 $W=kP$ ，求 k 的值。这称为椭圆曲线离散对数问题。

☒ 定义 2-4 椭圆曲线密钥算法

选椭圆曲线上的一点 G ，并选取一个保密数字 k 作为私钥，公钥为 G 和 P_A ，其中 $P_A=kG$ 。

☒ 定义 2-5 椭圆曲线加密算法

设明文为 m ，在曲线上找一点 P_m ，使其 x 坐标值与 y 坐标值之差为 m 。再选择一随机数 r 。则密文为

$$C = \{ rG, P_m + rP_A \} \quad (2-19)$$

定义 2-6 椭圆曲线解密算法

通过用私钥与第一点相乘，并减去第二点，得到明文 m ：

$$SP_m + rP_A - k(rG) = P_m + r(kG) - k(rG) = P_m \quad (2-20)$$

2.4 消息认证

2.4.1 消息认证码

消息认证是用来验证消息完整性的一种机制或服务。消息认证确保收到的数据确实和发送时的一样（即没有修改、插入、删除或重放），且发送方声称的身份是真实有效的。任何消息认证或数字签名机制在功能上基本可看作是有两层。下层中一定有某种产生认证符的函数，认证符是一个用来认证消息的值；上层协议中将该函数作为原语使接收方可以验证消息的真实性。用来做认证的函数主要有消息认证码和散列函数。

消息认证码（Message Authentication Code, MAC），是一种认证技术，它利用密钥来生成一个固定长度的短数据块，并将该数据块附加在消息之后。在这种方法中假定通信双方，比如 A 和 B，共享密钥 K 。若 A 向 B 发送消息，则 A 计算 MAC，它是消息和密钥的函数，即 $MAC=C(K, M)$ ，其中：

M ——输入消息；

C ——MAC 函数；

K ——共享的密钥；

MAC——消息认证码。

消息和 MAC 一起被发送给接收方。接收方对收到的消息用相同的密钥进行相同的计算得出新的 MAC，并将接收到的 MAC 与其计算出的 MAC 进行比较，如果假定只有收发双方知道密钥，那么若接收到的 MAC 与计算得出的 MAC 相等，则：

① 接收方可以相信消息未被修改。如果攻击者改变了消息，但无法改变相应的 MAC，所以接收方计算出的 MAC 将不等于接收到的 MAC。因为我们已假定攻击者不知道密钥，所以他不知道应如何改变 MAC 才能使其与修改后的消息相一致。

② 接收方可以相信消息来自真正的发送方。因为其他各方均不知道密钥，因此他们不能产生具有正确 MAC 的消息。

2.4.2 Hash 函数

单向散列（Hash）函数，又称杂凑函数，是消息认证码的一种变形。它是密码学的一个基本工具，在信息安全领域有广泛和重要的应用，主要作用是数据完整性验证和消息认证。

和消息认证码一样，散列函数的输入是可变大小的消息 M ，输出是固定大小的散列码 $H(M)$ 。与 MAC 不同的是，散列码并不使用密钥，它仅是输入消息的函数。散列码也称消息摘要。散列码是所有消息位的函数，它具有错误检测能力，即改变消息的任何一位或多位，都会导致散列码的改变。

散列值 h 由下述形式的函数 H 生成：

$$h=H(M)$$

其中， M 是一个变长消息， $H(M)$ 是定长的散列值。

散列函数本身并不提供保密，其目的是要产生消息的“指纹”。散列函数要能用于消息认证，它必须具有下列性质：

- ① H 可应用于任意大小的数据块；
- ② H 产生定长的输出；
- ③ 对任意给定的 x ，计算 $H(x)$ 比较容易，用软件和硬件均可实现；
- ④ 单向性，对任意给定的散列码 h ，找到满足 $H(x)=h$ 的 x 在计算上是不可行的；
- ⑤ 抗弱碰撞性，对任何给定的分组 x ，找到满足 $y \neq x$ 且 $H(x)=H(y)$ 的 y 在计算上是不可行的；
- ⑥ 抗强碰撞性：找到任何满足 $H(x)=H(y)$ 的偶对 (x,y) 在计算上是不可行的。

前 3 个条件是散列函数实际应用于消息认证中所必须满足的。

单向性对使用秘密值的认证技术极为重要。虽然该秘密值本身并不传送，但若散列函数不是单向的，则攻击者可以按照如下方式很容易地找出这个秘密值：若攻击者截获到传送的消息，则他可以得到消息 M 和散列码 $C = H(S_{AB} \parallel M)$ ，然后求出散列函数的逆，从而得出 $S_{AB} \parallel M = H^{-1}(C)$ 。由于攻击者已知 M 和 C ，所以可得出 S_{AB} 。

抗弱碰撞性可以保证，不能找到与给定消息具有相同散列值的另一消息，因此可以在使用散列码加密的方法中防止攻击者伪造。

抗强碰撞性用于抵生日攻击。

大多数重要的 Hash 函数都设计成了一个迭代过程，其处理过程如图 2-17 所示。首先对 Hash 函数的原始输入进行预处理，使之长度为 r 的整倍数，得到 $x = x_1x_2 \cdots x_t$ ， x_i 长度为 r ， $1 \leq i \leq t$ 。 $H_0 = IV$ 为初值， $H_i = f(x_i, H_{i-1})$ 。 f 是 Hash 函数的压缩函数， g 是输出变换。

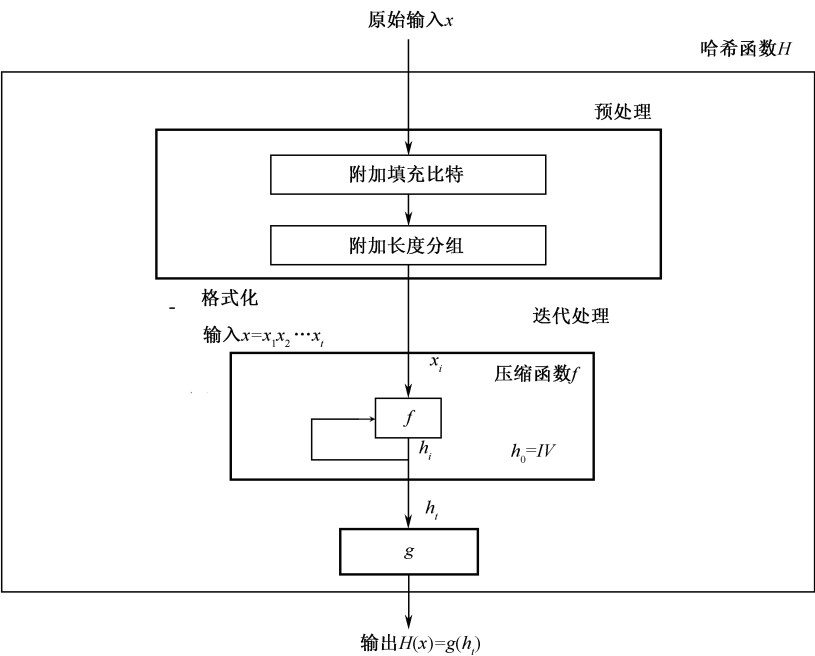


图 2-17 Hash函数处理过程

2.4.3 MD5 算法

1. MD5 算法的发展过程

简单的散列函数并没有提供用于数字签名的足够的安全性，人们已经为此目的建议了几个更复杂的函数，其中之一是 MD5 算法（Message-Digest Algorithm 5）^[25]，20 世纪 90 年代初由 Ronald L.Rivest 开发，经 MD2、MD3 和 MD4 算法发展而来。MD5 算法的前身是 MD4，1990 年 10 月作为 RFC 1320 提出，1992 年 4 月公布了 MD4 的改进（RFC 1321），称为 MD5。

MD5 是一种不可逆的算法，即对生成的密文求逆，对应着无穷个逆。它的作用是让大容量信息在用数字签名软件签署私人密钥前被“压缩”成一种保密的格式（即把一个任意长度的字节串变换成一定长的大整数）。不管是 MD2、MD4，还是 MD5，它们都需要获得一个随机长度的消息，并产生一个 128 位的消息摘要。虽然这些算法的结构或多或少有些相似，但是 MD2 的设计与 MD4 和 MD5 完全不同，因为 MD2 是为 8 位计算机做过设计优化的，而 MD4 和 MD5 却是面向 32 位的计算机的。这三个算法的描述和 C 语言源代码在 RFC 1321 中有详细的描述。

Ronald L.Rivest 在 1989 年开发出了 MD2 算法，在这个算法中，首先对消息进行数据补位，使消息的字节长度是 16 的倍数，然后，以一个 16 位的检验和追加到消息末尾，并且根据这个新产生的消息计算出散列值。后来，Rogier 和 Chauvaud 发现如果忽略了检验和将产生 MD2 冲突。

为了加强算法的安全性，Ronald L.Rivest 在 1990 年又开发出了 MD4 算法。MD4 算法同样需要填补消息以确保信息的字节长度加上 448 后能被 512 整除（消息字节长度 $\text{mod } 512 = 448$ ）。然后，一个以 64 位二进制数表示的消息的最初长度被添加进来。消息被处理成 512 位迭代结构的分组，而且每个分组要通过三个不同步骤的处理。Den Boer 和 Bosselaers 以及其他很快发现了攻击 MD4 版本中第一步和第三步的漏洞。Dobbertin 向大家演示了如何利用一台普通的 PC 在几分钟内找到 MD4 完整版本中的冲突（这个冲突实际上是一种漏洞，它将导致对不同的内容进行加密却可能得到相同的加密结果），毫无疑问，MD4 就此被淘汰掉了。

尽管 MD4 算法在安全上存在这么大的漏洞，但它对其后的安全散列算法却有着不可忽视的引导作用。除了 MD5 外，其中比较有名的还有 SHA-1、Snefru 以及 HAVAL 等。

一年以后，即 1991 年，Ronald L.Rivest 开发出了技术上更为成熟的 MD5 算法。

2. MD5 算法的处理过程

算法 2-6 MD5 算法

算法的输入为任意长的消息，以 512 位分组来处理输入的消息，每一分组又划分为 16 个 32 位子分组。算法的输出由 4 个 32 位分组组成，将它们级联形成一个 128 位散列值。

（1）消息填充

对消息进行填充，使消息长度比 512 的倍数少 64 位（这 64bit 用来表示消息长度），填充内容由一个 1 和后续的 0 组成。

例如，如果原消息长度为 1000 位，则要填充 472 位，使消息长度为 1472 位，因为 $64+1472=1536$ ，是 512 的倍数 ($1536=512*3$)。

(2) 附加消息的长度

最后一组的后 64 位用来表示消息长度 K 在 $\text{mod } 2^{64}$ 下的值。

消息长度的计算不包括填充位，例如，如果原消息为 1000 位，填充 472 位，其消息长度为 1000，而不是 1472。

经过处理后，得到长度为 512 位的一系列分组 Y_1, Y_2, \dots, Y_{L-1} 。

(3) 缓冲区初始化

算法使用 4 个 32 位的存储器 A、B、C、D 作为缓冲区，以存储中间结果和最终散列值。其初始值的十六进制表示为：A=0x01234567，B=0x89ABCDEF，C=0xFEDCBA98，D=0x76543210，低位字节在前 (表 2-7)。

表 2-7 缓冲区的数据存储形式

A	十六进制	01	23	45	67
B	十六进制	89	AB	CD	EF
C	十六进制	FE	DC	BA	98
D	十六进制	76	54	32	10

(4) 消息处理

消息处理过程是个循环，对消息中的多个 512 位分组进行运算，每个分组 $Y_i (i=0,1,\dots,L-1)$ 都经 H_{MD5} 函数处理， H_{MD5} 是算法的核心。

MD5 算法的处理过程如图 2-18 所示，图中 IV 是存储器 A、B、C、D 的初始值， $CV_i (i=1,\dots,L-1)$ 为存储的中间结果。

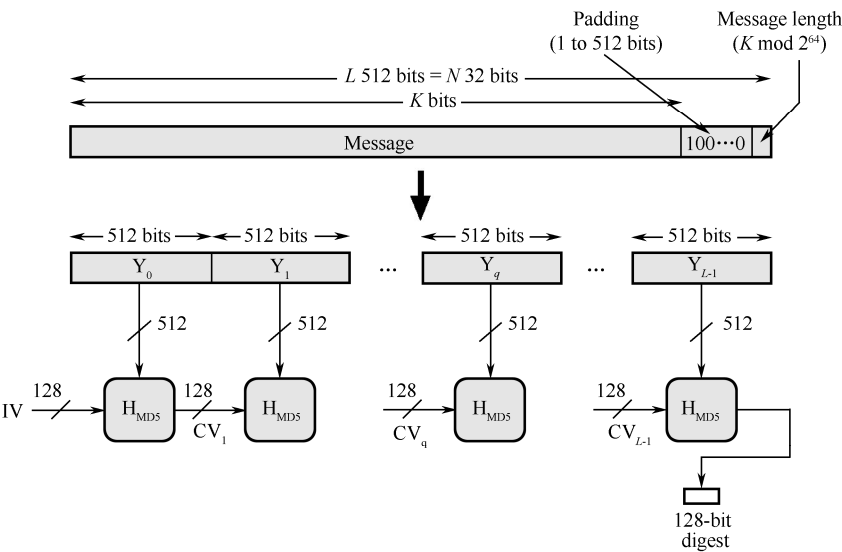


图 2-18 MD5 算法的处理过程

MD5 处理 512 位分组的过程，即 H_{MD5} 函数如图 2-19 所示。

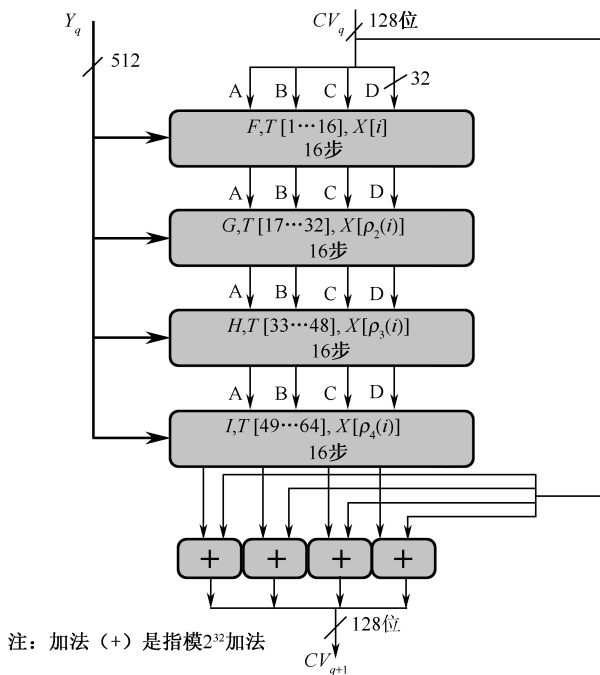


图 2-19 MD5 处理 512 位分组的过程

H_{MD5} 是算法的核心，有 4 轮处理过程，其结构一样，但所用的逻辑函数不同，分别表示为 F 、 G 、 H 、 I 。每轮的输入为当前处理的消息分组 Y_q 和缓冲区的当前值 A 、 B 、 C 、 D ，输出仍放在缓冲区中以产生新的 A 、 B 、 C 、 D 。每轮处理过程还需加上常数表 T 中四分之一一个元素，分别为 $T[1\cdots 16]$ 、 $T[17\cdots 32]$ 、 $T[33\cdots 48]$ 、 $T[49\cdots 64]$ 。第 4 轮的输出再与第 1 轮的输入 CV_q 相加，相加时将 CV_q 看做 4 个 32bit 的字，每个字与第 4 轮输出的对应的字按模 2^{32} 相加，相加的结果即为 H_{MD5} 的输出。

3. MD5 算法的安全性

MD5 在 MD4 的基础上增加了“安全-带子”(Safety-Belts)的概念。虽然 MD5 比 MD4 稍微慢一些，但却更安全。在 MD5 算法中，消息摘要的长度和填充的必要条件与 MD4 完全相同。Den Boer 和 Bosselaers 曾发现了 MD5 算法中的假冲突 (Pseudo-Collisions)。

Van Oorschot 和 Wiener 曾经考虑过一个在散列中强力搜寻冲突的函数 (Brute-Force Hash Function)，而且他们猜测一个被设计专门用来搜索 MD5 冲突的计算机 (这台计算机在 1994 年的制造成本大约是一百万美元) 可以平均每 24 天就找到一个冲突。

2004 年，中国数学家王小云证明 MD5 数字签名算法可以产生碰撞。

2007 年，Marc Stevens, Arjen K. Lenstra 和 Benne de Weger 进一步指出通过伪造软件签名，可重复性攻击 MD5 算法。

2008 年，荷兰埃因霍芬技术大学科学家成功把 2 个可执行文件进行了 MD5 碰撞。

MD5 碰撞使得两个不同的程序可以被计算出同一个 MD5 散列值。研究者使用前缀碰撞

法 (chosen-prefix collision), 使程序前端包含恶意程序, 利用后面的空间添上垃圾代码凑出同样的 MD5 散列值, 显然这样会为病毒大开方便之门。

由于 MD5 算法的使用不需要支付任何版权费用, 所以在一般的情况下 (非绝密应用领域, 即便是应用在绝密领域内, MD5 也不失为一种优秀的中间技术), MD5 是安全的。

2.4.4 SHA 算法

SHA (Secure Hash Algorithm, 安全散列算法) 是美国国家安全局 (National Security Agency, NSA) 设计, 美国国家标准与技术研究院 (NIST) 发布的一系列密码散列函数。SHA-0 于 1993 年发布 (PUBS 180-1-1993), 两年之后, SHA-1 发布 (FIPS PUBS 180-1-1995)。SHA-1^[26] 是目前国际通用的 Hash 函数算法, 被认为是现代网络安全的基石, 广泛使用于银行、安全通信以及电子商务中。但 2005 年 2 月 13 日, 王小云等人宣告破解了 SHA-1^[27], 此举成为破译 MD5 之后, 国际密码学领域的又一突破性研究成果。另外还有四种变体: SHA-224、SHA-256、SHA-384 和 SHA-512 (有时候也被称为 SHA-2)。

算法 2-7 SHA-1

算法的输入为小于 2^{64} 位长的任意消息, 分为 512 位长的分组, 输出为 160 位长的消息摘要。

(1) 消息填充

对输入的数据进行填充, 使其长度为 512 位的整数倍。由于最后要加上 64 位的原始消息长度, 所以要填充的消息长度 $\equiv (448 \bmod 512)$, 填充内容由一个 1 和后续的 0 组成。

例如, 在 8 位 ASCII 码系统中, 消息 “abc” 长度为 $8 \times 3 = 24$ 位, 需要填充 $448 - 24 = 424$ 位。最后加上 64 位原始消息长度二进制表示后, 就得到 512 位的填充消息:

$$\underbrace{01100001}_{a} \underbrace{10110001}_{b} \underbrace{1001100011}_{c} \overbrace{00 \cdots 0}^{423} \overbrace{00 \cdots 011000}^{64} \quad L=24$$

(2) 附加消息的长度

最后一组的后 64 位用无符号整数表示填充前消息的长度。经过这两步的处理, 得到长度为 512 位的一系列分组 Y_1, Y_2, \dots, Y_{L-1} , 扩展消息的长度为 $L \times 512$ 位。

(3) 缓冲区初始化

算法使用 5 个 32 位的存储器 A、B、C、D、E 作为缓冲区, 其初始值的十六进制表示为: $A=0x67452301, B=0xEFCDAB89, C=0x98BADCFE, D=10325476, E=0xC3D2E1F0$ 。

前四个与 MD5 相同, 但存储为高位字节在前的格式。

(4) 消息处理

算法的核心是被被称为压缩函数 (Compression Function) 的模块, 这个模块包括 4 次循环, 每次循环又包含 20 个处理步骤。4 次循环具有相似的结构, 但每次循环使用不同的基本逻辑函数 f_i 。

$$f_t(B, C, D) = (B \wedge C) \oplus (\sim B \wedge D) \quad (0 \leq t \leq 19)$$

$$f_t(B, C, D) = (B \oplus C \oplus D) \quad (20 \leq t \leq 39)$$

$$f_t(B, C, D) = (B \wedge C) \oplus (B \wedge D) \oplus (C \wedge D) \quad (0 \leq t \leq 19)$$

$$f_t(B, C, D) = (B \oplus C \oplus D) \quad (0 \leq t \leq 19)$$

其中, \wedge 是与运算, \sim 是非运算, \oplus 是异或运算。

压缩函数的结构如图 2-20 所示。

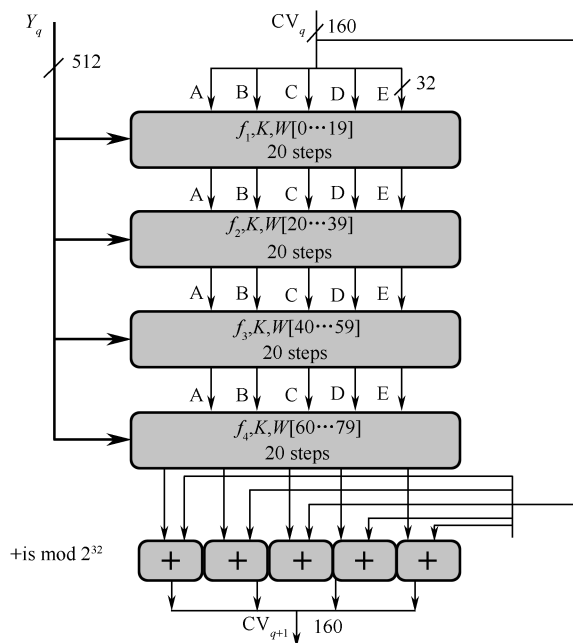


图 2-20 压缩函数示意图

其中:

$CV_0 = IV$, IV 是 $ABCDE$ 的初始值;

$$CV_{q+1} = \text{SUM}_{32}(CV_q, ABCDE_q)$$

SUM_{32} 表示对每一个输入对的字单独相加, 使用 $\text{mod } 2^{32}$ 加法;

$ABCDE_q$ 是对第 q 轮消息数据块处理最后所得的结果;

$MD = CV_L$, L 是数据块的个数, MD 是最后的消息摘要值。

SHA-1 一共需要 80 个 32 位的常量 K_t :

$$K_t = 0x5A827999 \quad (0 \leq t \leq 19)$$

$$K_t = 0x6ED9EBA1 \quad (20 \leq t \leq 39)$$

$$K_t = 0x8F1BBCDC \quad (40 \leq t \leq 59)$$

$$K_t = 0xCA62C1D6 \quad (60 \leq t \leq 79)$$

算法还用到 80 个字的缓冲区 $W_t(t=0,1,\cdots,79)$ 和 1 个字的 TEMP 缓冲区。每一轮中 20 步的每一步运算结构如图 2-21 所示。

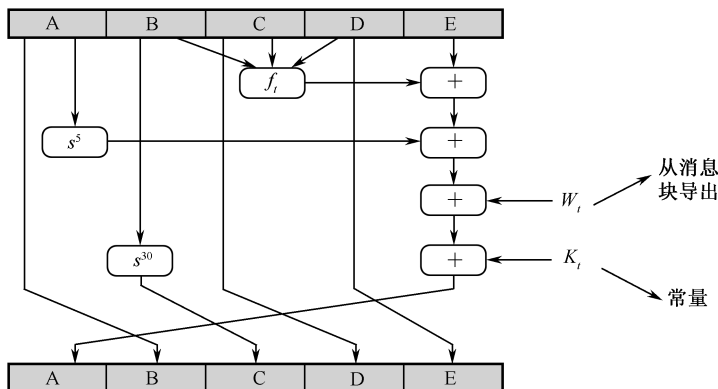


图 2-21 每一轮中 20 步的每一步运算结构

$$\text{TEMP} = S^5(A) + f_t(B, C, D) + E + W_t + K_t$$

$$E=D, \quad D=C, \quad C=E, \quad C=S^{30}(B), \quad B=A, \quad A=TEMP$$

其中, S^n 为位循环移位运算, $S^n(X) = (X \ll n) \text{OR} (X \gg 32 - n)$ 。

SHA-1 是由 MD5 算法演变而来的，它们之间最大的区别在于前者产生的摘要长度比后者长 32 位。SHA-1 对于强行攻击，产生任何一个消息使其摘要等于给定消息摘要的难度为 2^{160} 数量级的操作，较 MD5 算法的 2^{128} 数量级操作要难。因此，SHA-1 抵抗强力攻击的能力很强。同时，因为 SHA-1 的循环次数有 80 次，比 MD5 算法的 64 次要多，所以其运行速度也较慢。

2.5 数字签名

2.5.1 数字签名概述

数字签名的概念由 Diffie 和 Hellman 于 1976 年提出^[3], 是现代密码学最重要最基本的概念之一。尽管 Diffie 和 Hellman 在文献[3]中提出了用公钥密码实现数字签名的方法, 但他们并没有给出具体的数字签名方案。两年后, 即 1978 年, 由 Rivest, Shamir 和 Adleman 提出了基于大整数分解难题上的著名的 RSA 签名方案, 由于 RSA 签名算法简单、高效、便于应用, 目前国际上一些标准化组织如 ISO、ITU、SWIFT 等都把它作为标准采用, PGP (Pretty Good Privacy) 也采用 RSA 作为其签名算法。

经过多年的发展,已经有各种高效、安全的数字签名方案被提出,其中比较经典的有基于有限域离散对数问题的 ElGamal 数字签名方案^[5]、Schnorr 数字签名方案^[28]。1991 年美国 NIST 推出了数字签名算法标准——DSA/DSS^[7]。随着对数字签名研究的不断深入和电子商务的快速发展,出现了具有特殊性质的数字签名,如 1982 年 Chaum 引入了盲签名的概念^[9],1991

年 Chaum 和 Heyst 引入了群签名^[10]。

数字签名的设计思想等同于手写签名,即将签名者的身份与其签署的消息绑定,表示某人已对某消息进行了签字。任何的验证者均能验证消息确实为签名者所签署,而伪造一个合法用户的签名是困难的。数字签名是实现数字通信中可认证性、完整性和不可否认性的重要技术。

数字签名是一种认证机制,它使得消息的发送者可以添加一个起签名作用的码字。通过计算消息摘要并用发送者的私钥加密摘要值来生成签名。签名保证了消息的来源和完整性。

消息认证可以保护信息交换双方不受第三方的攻击,但是它不能处理通信双方自身发生的攻击。在收发双方不能完全信任的情况下,就需要除认证之外的其他方法来解决这些问题。数字签名是解决这个问题的最好方法,它的作用相当于手写签名。

数字签名必须具有下列特征:

- ① 能验证签名者的身份、签名的日期和时间;
- ② 能认证被签名的消息内容;
- ③ 数字签名能由第三方仲裁,以解决通信双方的争议。

因此,数字签名具有认证功能。

数字签名分为直接数字签名和仲裁数字签名。

直接数字签名只涉及通信双方。假定接收方已知发送方的公钥,则发送方可以通过用自己的私钥对整个消息或消息摘要加密来产生数字签名。直接数字签名有一个弱点,即这些方法的有效性依赖于发送方私钥的安全性。如果发送方想否认以前曾发送过某条消息,那么他可以称其私钥已丢失或被盗用,其他人伪造了他的签名,这种情况可以要求每条要签名的消息都包含一个时间戳,以及在密钥被泄密之后应立即向管理中心报告。另一种可能是, X 的私钥可能在时刻 T 被盗用,但攻击者可用 X 的签名签发一条消息并加盖一个在 T 或之前的时间戳。

仲裁数字签名可以解决直接数字签名中出现的问题。在这种类型的方法中,仲裁者起着关键的作用,通信各方都应非常信任仲裁机制。下面是一个仲裁数字签名的例子, X 表示发送方, Y 表示接收方, A 表示仲裁者, M 表示消息, T 表示时间戳。采用公钥算法进行签名和加密,仲裁者不能阅读消息:

- ① $X \rightarrow A: ID_x \parallel E(PR_x, [ID_x \parallel E(PU_y, E(PR_x, M))])$
- ② $A \rightarrow Y: E(PR_a, [ID_x \parallel E(PU_y, E(PR_x, M)) \parallel T])$

X 对消息 M 两次加密,即先用其私钥 PR_x 对消息 M 签名,然后再用 Y 的公钥 PU_y 加密,得到加密后的签名; X 再用 PR_x 对其标识和上述加密后的签名 $[ID_x \parallel E(PU_y, E(PR_x, M))]$ 进行签名,并连同 ID_x 一起发送给 A 。上述两层加密后的消息对仲裁者(以及除 Y 外的所有人)是秘密的,但是 A 可以对外层解密以验证消息确实发自 X (因为只有 X 有私钥 PR_x)。 A 检查 X 的公/私钥对是否有效,若是则消息是有效的,然后 A 再用其私钥 PR_a 对 ID_x 及两次加密后的消息和一个时间戳加密后传给 Y 。

2.5.2 数字签名标准

数字签名标准 (Digital Signature Standard, DSS) 是由美国 NIST 公布的联邦信息处理标准 FIPS 186, 其中采用了安全散列算法 (SHA), 给出了一种新的数字签名方法, 即数字签名算法 (Digital Signature Algorithm, DSA)。DSS 最初提出于 1991 年, 1993 年根据公众对其安全性的反馈意见进行了一些修改, NIST 于 1994 年 5 月出版了 FIPS PUB 186^[7], 1996 年又稍做修改。2000 年发布了该标准的扩充版, 即 FIPS PUB 186-2。最新版本还包括基于 RSA 和椭圆曲线密码的数字签名算法。

DSA 建立在求离散对数的困难性及 ElGamal^[5]和 Schnorr^[28]最初提出的方法之上。

算法 2-8 DSA 密钥生成算法

(1) 全局公钥 (p, q, g)

p : 满足 $2^{L-1} < p < 2^L$ 的大素数, 其中 $512 \leq L \leq 1024$, 且 L 是 64 的倍数。

q : $p-1$ 的素因子, 满足 $2^{159} < q < 2^{160}$, 即 q 长度为 160 位。

g : $g \equiv h^{(p-1)/q} \pmod{p}$, 且 $1 < h < (p-1)$, 使 $h^{(p-1)/q} \pmod{p} > 1$ 。

(2) 用户的私钥 x

x 为满足 $0 < x < q$ 的随机数或伪随机数。

(3) 用户的公钥 y

$$y \equiv g^x \pmod{p} \quad (2-21)$$

(4) 用户为每个待签消息选取的秘密数 k

k 是满足 $0 < k < q$ 的随机数或伪随机数。

算法 2-9 DSA 签名算法 (图 2-22 (a))

对消息 $M \in Z_p^*$, 其签名为:

$$S = \text{Sig}_k(M, k) = (r, s)$$

其中, $S \in Z_p \times Z_p$,

$$r \equiv (g^k \pmod{p}) \pmod{q} \quad (2-22)$$

$$s \equiv (k^{-1}(h(M) + xr)) \pmod{q} \quad (2-23)$$

$h(M)$ 是由 SHA 求出的杂凑值。

算法 2-10 DSA 验证算法 (图 2-22 (b))

设接收方收到的消息为 M , 签字为 (r, s) 。计算

$$\omega \equiv s^{-1} \pmod{q}, u_1 \equiv (h(M)\omega) \pmod{q} \quad (2-24)$$

$$u_2 \equiv r\omega \pmod{q}, v \equiv ((g^{u_1} y^{u_2}) \pmod{p}) \pmod{q} \quad (2-25)$$

$$\text{Ver}(M, r, s) = \text{Ture} \Leftrightarrow v = r \quad (2-26)$$

接收端的验证依赖于 r , 但是 r 却根本不依赖于消息, 它是 k 和全局公钥的函数。 k 模 p 的乘法逆元传给函数 f_1 , f_1 的输入还包含消息的 Hash 码和用户私钥。函数的这种结构使接收

方可利用其收到的消息和签名、它的公钥以及全局公钥来恢复 r 。

由于求离散对数的困难性，攻击者从 r 恢复出 k 或从 s 恢复出 x 都是不可行的。

另外，产生签名中需要进行复杂指数运算 $g^k \bmod p$ ，但由于它不依赖于被签名的消息，因此可以预先计算。实际上，用户甚至可以根据需要预先计算许多个用于签名的 r 。

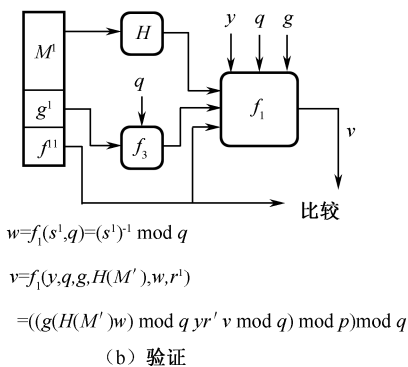
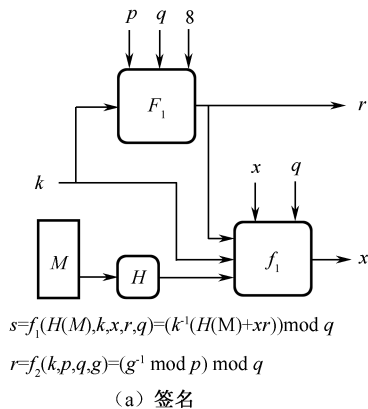


图 2-22 DSA 签名和验证

2.6 特殊的数字签名

2.6.1 盲签名

传统数字签名的一个基本特征是签名者知道所签消息的内容。但在某些特殊情况下，人们并不希望这样，盲签名正是这样一种特殊的签名。1982 年，D. Chaum 首先提出了盲签名的概念。D. Chaum 对盲签名曾经给出了一个非常直观的说明：所谓盲签名，就是先将要隐蔽的文件放进信封里，当文件在一个信封中时，任何人都不能读它。对文件签名就是通过在信封里放一张复写纸，当签名者在信封上签名时，他的签名便透过复写纸签到了文件上。下面给出盲签名的数学定义。

☒ 定义 2-7 盲签名

一个盲签名方案是一个算法的三元组 $(\text{gen}, \text{IP}, \text{Ver})$ ，其中 gen 是概率型算法， IP 是签名者 S 和接收者 R 之间的交互协议 (R, S) ， Ver 是确定型验证算法。 gen 输入系统参数，输出签名者 S 的私钥 x 和公钥 y 。协议 (R, S) 中 S 的输入是 x ， R 的输入是盲化后的消息 \tilde{m} 和 y 。 R 的输出是 m 的签名 σ_x 。 Ver 输入消息 m 、签名 σ_x 和 S 的公钥 y ，输出 true 或 false。

一个盲签名方案不仅保留有数字签名的各类特性，而且还拥有以下一些特殊的性质。

- ① 正确性：如果 R 和 S 都正确地执行 IP ，则签名满足 Ver 。
- ② 不可伪造性：任何不知道私钥 x 的人都不可能产生正确的签名 σ_x 。
- ③ 盲性：签名者不知道他所签署消息的具体内容。

④ 不可追踪性: 签名者仅知盲消息 \tilde{m} 的签名 $\text{sig}(\tilde{m})$, 而不知真实消息 m 的签名 $\text{sig}(m)$ 。当签名消息被公布后, 签名者无法知道这是他何时签署的, 即签名者不能把签名和盲消息对应起来。

1982 年, D. Chaum 在 CRYPTO 会议上发表了“Blind signature for untraceable payments”一文^[9], 提出了一个基于 RSA 公钥密码系统的盲签名方案。该方案是第一个盲签名方案, 它主要用于电子选举系统和电子支付系统中保护用户的匿名性。该方案是目前性能最好的一个方案, 大多数电子货币系统和电子投票系统的设计都采用此方案。盲签名方案比一般的数字签名方案多了一个盲化消息和去盲化的过程, 从而保证消息不会被签名者知道。首先接收者将待签名的消息进行盲变换, 把变换后的盲消息发给签名者, 经签名者签名后再发回给接收者, 接收者对签名进行逆盲变换, 得出的便是签名者对原始消息的盲签名, 即: 消息 \rightarrow 盲变换 \rightarrow 签名 \rightarrow 接收者 \rightarrow 逆盲变换。签名协议如下。

1. 初始化阶段

签名者 B 取两个大素数 $p, q, n = p \cdot q, \varphi(n) = (p-1)(q-1)$, 随机选取 e 满足 $1 < e < \varphi(n)$ 且 $\text{gcd}(e, n)=1$, 计算 d 满足 $1 < d < \varphi(n)$ 且 $de \equiv 1 \pmod{\varphi(n)}$, B 的公钥是 (n, e) , 私钥是 d , $h()$ 是具有无碰撞性的散列函数。

2. 签名阶段

步骤 1: 消息拥有者 A 随机选取与 n 互素的 k , 计算 $\tilde{m} = h(m) \cdot k^e \pmod{n}$, 并将 \tilde{m} 发送给签名者 B。

步骤 2: B 接收到 \tilde{m} 后, 计算 $\tilde{s} = \tilde{m}^d \pmod{n}$, 并将 \tilde{s} 发送给 A。

步骤 3: A 接收到 \tilde{s} 后, 计算 $s = k^{-1}\tilde{s} \pmod{n}$, s 是 B 对 $h(m)$ 的签名, 即 $s = h(m)^d \pmod{n}$ 。

3. 验证阶段

计算 $h'(m) = s^e \pmod{n}$, 若 $h'(m) = h(m)$ 成立, 则验证了 s 是 B 对 $h(m)$ 的盲签名, 否则拒绝接受。

2.6.2 群签名

群签名是 D. Chaum 和 E. Van. Heyst 在 1991 年 EURCRYPT'91 上提出的一种特殊签名方案^[10]。这个方案的提出主要针对如下问题: 一个公司的几台计算机都连接到局域网上, 各个部门都有自己的打印机, 且只有本部门的人员才允许使用自己的打印机。因此在打印之前必须验证用户是否在本部门工作。同时为了保密需要, 公司不想暴露用户姓名。但是下班时如果发现打印机使用很频繁, 主管经理必须指出是谁滥用了那台打印机。

群签名具有以下几个特性。

- ① 只有群成员才能生成合法签名。
- ② 签名验证者可以证实签名的真伪, 但是无法知道签名者的身份。
- ③ 如果发生争议可以由一个群管理员“打开”签名确定签名者的身份。

由于群签名有以上特点,因此在电子商务、电子银行、电子投票、电子拍卖等领域有着广泛的应用前景。

D. Chaum 和 E. Van. Heyst 在首次提出群签名方案的概念时,也提出了四个群签名方案,尽管其中三个方案在打开签名时需要群成员协助,两个方案在系统建立后不能增加新成员,但它为群签名发展奠定了坚实基础。

Popescu 对 D. Chaum 和 E. Van. Heyst 提出的群签名方案进行了完善^[29],群签名方案可以看成包含如下的五个协议或算法的数字签名方案。

① 创建算法 (setup)。通过输入一个随机数,能够产生群的公开密钥 Y 和群管理员的秘密密钥 S 。

② 注册协议 (join)。这是群管理员与群成员之间的一个概率交互协议,可使得某个用户注册成为这个新的群成员。输出为一个群成员的身份证书以及一个只有群成员知道的秘密密钥。

③ 签名算法 (sign)。这是群成员与签名接收者之间的协议。当输入一个消息 M 与某个群成员的身份证书和秘密密钥后,输出消息 M 的签名。

④ 验证算法 (verify)。

这是一个确定性算法。当输入消息 M 和消息的签名以及群的公开密钥 Y 后,输出关于这个签名是否有效的判定。

⑤ 打开算法 (open)。这是一个确定性算法。当输入消息 M 和它的签名以及群管理员和秘密密钥 S 后,输出签名者的身份。

J. Camenisch 和 M. Stadler 在文献[30]中提出了一种群签名方案,该方案使用了零知识证明的概念。零知识证明可使证明方证明其拥有某些秘密值的知识,而不泄露有关秘密值的任何信息。零知识证明的思想使群签名方案发生了质的飞跃,对群签名的发展产生了重要的影响。

下面介绍该签名方案。为了简单起见,我们使用文献[30]群签名方案中的记号,不再赘述。

1. 系统建立

群管理员计算下列值:

- ① 一个 RSA 模 n 及两个公开的指数 $e_1, e_2 > 1$, 并且 $\gcd(e_2, \phi(n)) = 1$ 。
- ② 两个整数 $f_1, f_2 > 1$, 使得在不知道 n 的因子分解时计算其 e_1 次根及 e_2 次根是困难的。
- ③ 阶为 n 的循环群 $G = \langle g \rangle$, 使得在 G 中计算离散对数是困难的。
- ④ 一个元素 $h \in G$, 使得计算 h 关于 g 的离散对数是困难的。
- ⑤ 任选一个随机数 $\rho \in Z_n$, 令 $y_R = h^\rho$ 为群管理员的公钥。

群组的公钥 $Y = (n, e_1, e_2, f_1, f_2, G, g, h, y_R)$, 而 ρ 和 n 的素因子为群管理员的私钥。

2. 成员加入

为了成为一个群成员, Alice 首先计算她的成员私钥, 任选一个 $x \in_R Z_n^*$, 令 $y = x^{e_1} \pmod n$ 。Alice 保密 y 和 x 作为她的成员身份私钥。然后 Alice 计算 $z = g^y$, 公开 z 并以 z 代表她的身份, z 是 Alice 身份的公钥。在该方案中, 证书的形式为

$$v = (f_1 y + f_2)^{1/e_2} \pmod n \quad (2-27)$$

为了成为群成员, Alice 必须向群管理员注册这些值, 并获得成员证书。Alice 不能将 y 直接发送给群管理员, 否则群管理员可能冒充 Alice。因此, Alice 必须利用盲 RSA 签名提交。Alice 计算:

$$\tilde{y} = r^{e_2} (f_1 y + f_2) \pmod{n} \quad (2-28)$$

其中, $r \in_R Z_n^*$ 。

$$U = E - \text{SKROOTLOG}[\alpha : z = g^{\alpha^{e_1}}](m) \quad (2-29)$$

$$V = E - \text{SKROOTLOG}[\beta : g^{\tilde{y}} = (z^{f_1} g^{f_2})^{\beta^{e_2}}](m) \quad (2-30)$$

Alice 将 $\{z, \tilde{y}, U, V\}$ 发送给群管理员。群管理员收到 $\{z, \tilde{y}, U, V\}$ 后, 验证 $\{U, V\}$ 的正确性, 若正确, 群管理员则计算盲化的证书 $\tilde{v} = \tilde{y}^{1/e_2} \pmod{n}$, 并将 \tilde{v} 发送给 Alice。

Alice 去掉盲化因子 r , 可得到其成员身份证书:

$$v = \tilde{v} / r = (f_1 y + f_2)^{1/e_2} \pmod{n} \quad (2-31)$$

签名 U 说明 Alice 确实知道私钥, V 说明 Alice 正确盲化了 $(f_1 y + f_2)$ 。

3. 签名过程

为了代表群组对消息 m 签名, Alice 计算对消息的知识签名, 证明她是群组的注册成员。同时, 利用群管理员的公钥对其成员公钥进行加密, 这样使群管理员在必要的时候打开签名。具体实现为: Alice 任选一个随机数 $r \in_R Z_n^*$, 计算:

$$\textcircled{1} \quad \tilde{z} = h^r g^y, \quad r \in_R Z_n^* \quad (2-32)$$

$$\textcircled{2} \quad d = y_r^\gamma \quad (2-33)$$

$$\textcircled{3} \quad V_1 = E - \text{SKROOTREP}[(\alpha, \beta) : \tilde{z} = h^\alpha g^{\beta^{e_1}}](m) \quad (2-34)$$

$$\textcircled{4} \quad V_2 = E - \text{SKROOTREP}[(\gamma, \delta) : \tilde{z}^{f_1} g^{f_2} = h^\gamma g^{\delta^{e_2}}](m) \quad (2-35)$$

$$\textcircled{5} \quad V_3 = \text{SKREP}[(\varepsilon, \xi) : d = y_r^\varepsilon \wedge \tilde{z} = h^\varepsilon g^\xi](m) \quad (2-36)$$

则 Alice 对信息 m 的签名为 $(\tilde{z}, d, V_1, V_2, V_3)$ 。

4. 签名验证

Alice 通过验证 (V_1, V_2, V_3) 同时成立, 来证明对 m 的签名 $(\tilde{z}, d, V_1, V_2, V_3)$ 的正确性。通过对 (V_1, V_2, V_3) 的正确性验证可使验证者确信

$$\gamma \equiv \alpha \pmod{n}, \quad \delta^{e_2} = f_1 \beta^{e_1} + f_2 \pmod{n} \quad (2-37)$$

式 (2-37) 中的第 2 个等式表示 Alice 拥有成员证书 $v = \delta$, 而且其成员私钥为 $x = \beta$ 。通过对 V_3 的验证, 验证者确信 \tilde{z} 与 d 的计算使用了同一个随机数 $\gamma = \varepsilon$, 即 (d, \tilde{z}) 是 Alice 利用群管理员公钥 (h, y_r) 对成员公钥 z 的一个 ElGamal 加密。 V_3 的正确性确保了当需要时, 签名可以被群管理员打开。

5. 打开算法

当发生纠纷时, 群管理员打开 m 的签名 $(\tilde{z}, d, V_1, V_2, V_3)$ 。计算 $z = \tilde{z} / d^{1/\rho}$, 得到 Alice 的公钥 z 。为了证明 Alice 的公钥的确是用 \tilde{z} 和 d 加密而成的, 群管理员计算 $\text{SKREP}\{\alpha : \tilde{z} = z d^\alpha \wedge h = y_r^\alpha\}(m)$ 作为他判决的证据。

在此群签名方案中，为了计算群成员的身份证书，采用了盲 RSA 签名方案，群管理员不知道群成员的私钥，因此群管理员无法伪造群成员的签名。

2.7 PKI 认证体系

本节将对公钥基础设施（Public Key Infrastructure, PKI）认证体系所涉及的内容，如 PKI 的概念、PKI 的组成和 PKI 的标准等做简单的介绍。

2.7.1 PKI 的概念

PKI 是一种具有普适性的安全基础设施，它采用了证书管理公钥，通过第三方的可信任机构认证中心，把用户的公钥和用户的其他标识信息捆绑在一起，在 Internet 上验证用户的身份，保证网上数据的安全传输。

PKIX（Public Key Infrastructure Using X.509）工作组给 PKI 的定义为：“是一组建立在公开密钥算法基础上的硬件、软件、人员和应用程序的集合，它应具备产生、管理、存储、分发和废止证书的能力”^[31]。PKI 的主要目的是，通过自动管理密钥和证书，为用户建立起一个安全的通信信任机制，使用户可以在多种应用环境下方便地使用加密和数字签名技术。

PKI 的最基本元素是数字证书，所有安全操作主要通过数字证书来实现。数字证书是一个防篡改的数据集合，它包含有用户名、公开密钥以及用户的其他身份信息，可以证实一个公钥与某一用户身份之间的关系。而核心的实施者是认证中心 CA，是 PKI 中不可缺少的一部分，具有权威性，是一个普遍可信的第三方，主要向用户颁发数字证书。PKI 体制的基本原理是利用“数字证书”这一静态的电子文件来实施公钥认证。

PKI 提供了三个核心的安全服务。

- ① 认证：向一个实体确认另一个实体确实是他自己。
- ② 完整性：向一个实体确保数据没有被有意或者无意地修改。
- ③ 机密性：向一个实体确保除了接收者，无人能读懂数据的关键部分。

一个有效的 PKI 系统必须是安全和透明的，用户在获得加密和数字签名服务时，不需要详细了解 PKI 是怎样管理证书和密钥的。一个典型、完整、有效的 PKI 应用系统必须能够实现如下功能：注册、发证、密钥恢复、密钥产生、密钥更新、交叉认证和证书废止。

2.7.2 PKI 的组成

PKI 提供了实际实施和运作使用证书的系统所需的组件和服务，主要由以下几个部分组成^[32]。

1. CA（Certificate Authority，认证中心）

这是证书的签发机构，是 PKI 的核心，是 PKI 应用中权威的、可信任的、公正的第三方机构。CA 对任何一个主体的公钥进行公证，通过签发证书，将主体与公钥进行捆绑，负责确认身份和创建数字证书。以建立一个身份和一对公/私钥间的联系。

2. RA（Registration Authority，注册中心）

这是认证中心（CA）的延伸部分，与 CA 在逻辑上是一个整体。它本身并不签发证书，只负责接收用户的注册和申请鉴别，审核用户的身份，并决定是否同意 CA 给申请者签发数字证书。

3. 证书库

这是证书的集中存放地，提供公众查询。证书库可以是关系数据库，也可以是目录（目录服务器）、响应器等。通常用做 PKI 组成部分的证书库是目录，有时是 X.500 的目录，更常见的是 LDAP（Light Directory Access Protocol，轻量目录访问协议）目录，LDAP 实际上是对目录中信息的访问方法和协议的描述。

4. 密钥备份与恢复系统

对用户的解密密钥进行备份。如果用户的解密密钥丢失，则密文无法解密，造成数据丢失。密钥的备份与恢复应由可信机构来完成，只能针对解密密钥，签名私钥不能备份。

5. 证书撤销处理系统

由于某种原因，证书在有效期内需要作废、终止使用时，通过证书撤销列表（Certificate Revocation List，CRL）来实现，CRL 一般存放在目录系统中。

6. API（应用接口系统）

API 为各种各样的应用提供安全、一致、可信任的方式与 PKI 交互，确保所建立起来的网络环境安全可靠并降低管理成本。

完整的 PKI 包括认证政策的制定（包括遵循的技术标准、各 CA 之间的上下级或同级关系、安全策略、安全程度、服务对象、管理原则和框架等）、认证规则、运作制度的规定、所涉及的各方法律关系内容以及技术的实现等。

2.7.3 PKI 的标准

从整个 PKI 体系建立与发展的历程来看，与 PKI 相关的标准主要包括以下内容^[33]。

1. X.509

这是国际电信联盟（ITU-T）部分标准和国际标准化组织（ISO）的证书格式标准，X.509 定义了公钥证书结构的基本标准。X.509 的最初版本公布于 1988 年，当前版本是 X.509v3，

X.509v3 证书包括一组按预定义顺序排列的强制字段，还有可选扩展字段。可选扩展字段的加入，极大地增进了证书的灵活性。

2. X.500

X.500 是一套已经被国际标准化组织（ISO）接受的目录服务系统标准。X.500 目录服务是一个高度复杂的信息存储机制，包括客户机-目录服务器访问协议、服务器-服务器通信协议、完全或部分的目录数据复制、服务器链对查询的响应、复杂搜寻的过滤功能等。X.500 被认为是实现目录服务的最佳途径，但 X.500 的实现需要较大的投资，并且比其他方式速度慢，而其优势具有信息模型、多功能和开放性。

3. PKIX

PKIX（Public Key Infrastructure X.509）工作组成立于 1995 年，旨在开发必需的互联网标准来支持可互操作的 PKI。PKIX 的章程中包含了 4 项专门领域：

- ① 证书和证书户撤销列表概貌；
- ② 证书管理协议；
- ③ 证书操作协议；
- ④ 证书策略（Certification Policy, CP）和认证业务声明（Certification Practice Statement, CPS）结构。

4. PKCS

PKCS（The Public-Key Cryptography Standards）标准是由美国 RSA 数据安全公司及其合作伙伴制定的一组公钥密码学标准。PKCS 系列主要标准如下。

- ① PKCS#1 RSA 加密标准：定义 RSA 公开密钥算法加密和签名机制。
- ② PKCS#3 Diffie-Hellman 密钥协议标准：定义 Diffie-Hellman 密钥交换协议。
- ③ PKCS#5 基于口令的加密标准：描述由口令派生出来的安全密钥加密字符串的方法。
- ④ PKCS#6 扩展证书语法标准：定义了提供附加实体信息的 X.509 证书属性扩展的语法。
- ⑤ PKCS#7 密码消息语法标准：定义了一种通用的消息语法，包括数字签名和加密等用于增强的加密机制。
- ⑥ PKCS#8 私钥信息语法标准：定义了私钥信息语法和加密私钥语法。
- ⑦ PKCS#9 可选属性类型：定义一些用于 PKCS#6 证书扩展、PKCS#7 数字签名和 PKCS#8 私钥加密信息的属性类型。
- ⑧ PKCS#10 证书请求语法标准：描述证书请求的语法。
- ⑨ PKCS#11 密码令牌接口标准：“Cryptoki”，定义了一套独立于技术的程序设计接口，用于智能卡和 PCMCIA 卡之类的加密设备。
- ⑩ PKCS#12 个人信息交换语法标准：定义了个人身份信息（包括私钥、证书、各种秘密和扩展字段）的格式。

5. LDAP

轻量级目录访问协议（LDAP）不但简化了烦琐的 X.500 目录访问协议，并且在功能性、

数据表示、编码和传输方面都进行了相应的修改。1997 年，LDAP 第 3 版本成为了互联网标准。目前，LDAPv3 已经在 PKI 体系中被广泛应用于证书信息发布、CRL 信息发布、CA 政策以及与信息发布相关的各个方面。

目前 PKI 体系中已经包含了众多的标准和标准协议，随着 PKI 技术的不断进步和完善，将来还会有更多的标准和协议加入。

2.7.4 认证中心

认证中心（CA）是 PKI 的核心组成部分，本节将对 CA 的功能和结构做详细的描述。

1. CA 的概念和功能

CA 是 PKI 框架中唯一能够发布和撤销证书的实体，作为受信任的第三方，负责产生、分配并管理用户的数字证书，承担着 PKI 中公钥合法性检验的责任。它为每个使用公钥的用户发放基于数字签名的数字证书，用来表明证书中列出的用户名称与证书库中列出的公钥相对应。

CA 作为 PKI 系统的核心部分，直接为最终用户（个人、团体、设备等）签发支持各种应用的证书，并负责管理所签发的这些证书。CA 的核心功能就是发放和管理数字证书，主要包括证书申请、证书审批、证书颁发、证书查询、证书更新和证书撤销，同时也包括对证书撤销列表的管理功能。证书申请的方式分为在线申请和离线申请，用户可以申请各种功能的证书，包括 Web 浏览器证书、安全 E-mail 证书、Web 服务器证书和 CA 证书等。证书审批在 PKI 系统中是由注册中心（RA）来完成的，证书的生成由 CA 管理员在 CA 系统后台完成，证书信息存入证书库中。证书颁发的过程就是用户获得证书的途径，证书查询是指用户可以查询自己拥有的证书状态和详细信息，证书更新和证书撤销是 CA 系统和用户对证书的操作和维护。

2. CA 的结构

一个典型的 CA 系统包括安全服务器、RA 服务器、CA 服务器、LDAP 服务器和数据库服务器等（图 2-23）^[32]。

（1）安全服务器

安全服务器面向普通用户，提供证书申请、浏览、撤销和下载等安全服务。安全服务器与用户的通信采取安全信道方式（如 SSL 方式），用户首先得到安全服务器的证书（由 CA 颁发），然后用户与服务器之间的所有通信，包括用户填写的申请信息以及浏览器生成的公钥均以安全服务器的密钥进行加密传输，只有安全服务器利用自己的私钥解密才能得到明文，这样可以防止其他人通过窃听得到明文，从而保证了证书申请和传输过程中的信息安全性。

（2）RA 服务器

这是注册机构。登记中心服务器面向登记中心操作员，在 CA 体系结构中起承上启下的作用，一方面向 CA 转发安全服务器传输过来的证书申请请求，另一方面向 LDAP 服务器和安全服务器转发 CA 颁发的数字证书和证书撤销列表。在某些小型的认证中心中，RA 服务器

可与 CA 服务器合并，所有 RA 服务器的功能可由 CA 服务器来实现。

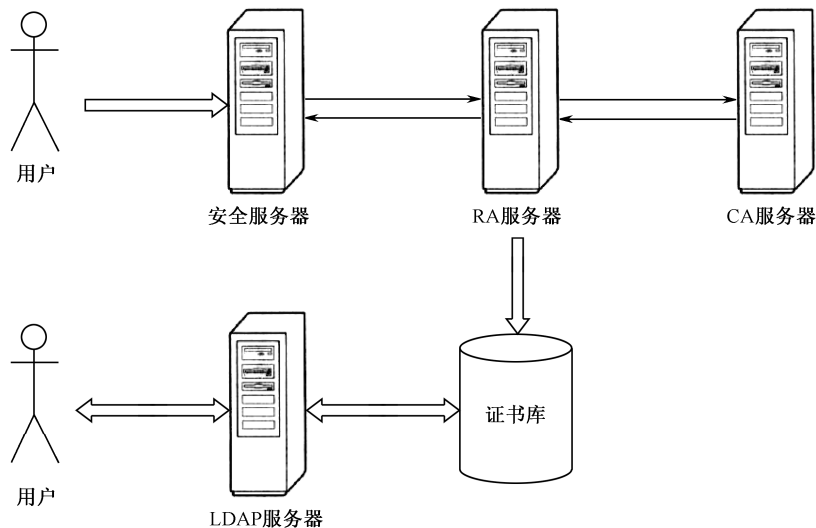


图 2-23 CA 的总体结构

（3）CA 服务器

CA 服务器是整个认证中心的核心，负责证书的签发。CA 首先产生自身的私钥和公钥（密钥长度至少为 1024 位），然后生成数字证书，并且将数字证书传输给安全服务器。CA 还负责为操作员、安全服务器以及注册机构服务器生成数字证书。操作员的数字证书需要转化为 PKCS#12 的格式，通过硬拷贝的方式嵌入登记中心操作员的浏览器中。安全服务器的数字证书和私钥也需要传输给安全服务器。CA 服务器是整个结构中最为核心的部分，存有 CA 的私钥以及发行证书的脚本文件，出于安全的考虑，应将 CA 服务器与其他服务器隔离，任何通信采用人工干预的方式，确保认证中心的安全。

（4）LDAP 服务器

LDAP 服务器提供目录浏览服务，负责将注册机构服务器传输过来的用户信息以及数字证书加入服务器。这样其他用户通过访问 LDAP 服务器就能够得到其他用户的数字证书了。

（5）数据库服务器

数据库服务器是认证机构中的核心部分，用于认证机构中数据（如密钥和用户信息等）、日志和统计信息的存储和管理。实际的数据库系统应采用多种措施，如磁盘阵列、双机备份和多处理器等方式，以维护数据库系统的安全性、稳定性、可伸缩性和高性能。

2.7.5 数字证书

公钥基础设施（PKI）最主要的任务是确立可信赖的数字身份，数字证书就提供了一种在网上验证身份的方式，本节将对数字证书的内容进行阐述。

1. 数字证书的概念

数字证书是一段包含用户身份信息、用户公钥信息以及身份验证机构数字签名的数据。

身份验证机构的数字签名可以确保证书信息的真实性，用户公钥信息可以保证数字信息传输的完整性，用户的数字签名可以保证数字信息的不可否认性。

数字证书是一个经认证中心（CA）数字签名的包含公开密钥拥有者信息以及公开密钥的文件，其作用类似于现实生活中的身份证，由权威的第三方机构认证中心（CA）发行，在一个身份和该身份的持有者拥有的公/私密钥对之间建立一种联系，如果用户想要确立身份，那么可以信赖一个特定的颁发机构，由这个机构根据用户要达到的目的来确立，然后生成一份可以证实该用户已经获得了有效身份的文件，而数字证书就是这样一种文件的电子形式。有了数字证书，就可以使别人相信该用户就是该身份的合法持有者。认证中心颁发的数字证书均遵循 X.509v3 标准。X.509 标准在编排公共密钥密码格式方面已被广为接受。X.509 证书已应用于许多网络安全领域，包括 IPSec（IP 安全）、SSL、SET、S/MIME。

2. X.509v3 版本的证书结构

在多数场合下，最广泛接受的证书格式是 X.509 标准，使用最多的就是 X.509v3 标准。图 2-24 给出了 X.509v3 证书结构。证书结构一般是通过 ASN.1（Abstract Syntax Notation One，抽象语符号）来描述和表示的。

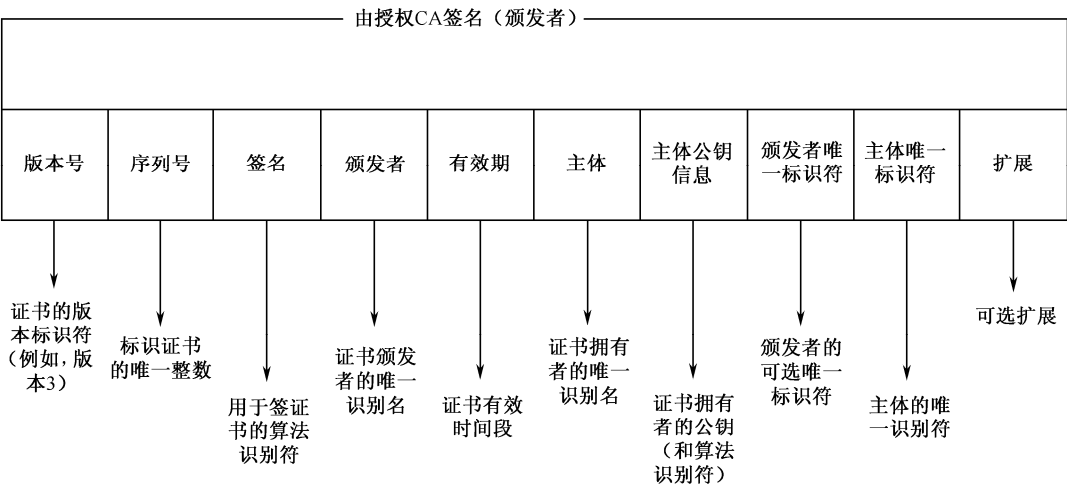


图 2-24 X.509v3 证书结构

- ① 版本号：标示证书的版本（如版本 1、版本 2 或版本 3）。
- ② 序列号：由证书颁发者分配的本证书的唯一标识符。
- ③ 签名：签名算法标识符（由对象标识符加上相关参数组成），用于说明本证书所用的数字签名算法。例如，SHA-1 和 RSA 的对象标识符就用来说明该数字签名利用 RSA 对 SHA-1 杂凑加密。
- ④ 颁发者：证书颁发者的可识别名（Distinguish Name, DN），这是必须说明的。
- ⑤ 有效期：证书有效的时间段。本字段由“Not Valid Before”和“Not Valid After”两项组成，它们分别由 UTC（Universal Time Coordinated，协调世界时，又称世界标准时间）或一般的时间标示（在 RFC 2459 中有详细的时间表示规则）。

- ⑥ 主体：证书拥有者的可识别名，此字段必须是非空的，除非使用了其他形式的名字。
- ⑦ 主体公钥信息：主体的公钥（以及算法标识符），这是必须说明的。
- ⑧ 颁发者唯一标识符：证书颁发者的唯一标识符，仅在版本 2 和版本 3 主体目录属性中要求，属于可选项，该字段在实际应用中很少使用，并且不被 RFC 2459 推荐使用。
- ⑨ 主体唯一标识符：证书颁发者的唯一标识符，仅在版本 2 和版本 3 中要求，属于可选项，该字段在实际应用中很少使用，并且不被 RFC 2459 推荐使用。
- ⑩ 扩展：可选的标注和专用扩展（仅在版本 3 里使用），它们包括 Authority 密钥标识符、主体密钥标识符、密钥使用、扩展密钥使用、CRL 分布点、私钥试用期、证书策略、策略映射、主题别名、颁发者别名和主体目录属性等。

2.8 DRM 加密

数字多媒体的使用为 DRM 加密技术带来了新的挑战，为了适应新的应用要求，产生了很多新的算法。文献[12]介绍了一些相关的文献，概述了目前 DRM 加密技术研究的主要三个方向，即 DRM 加密结构、加密算法和检验加密效果的方法，并分别介绍了三个方向中目前的研究成果。

2.8.1 DRM 加密概述

DRM 系统的目的是保护版权和数字多媒体资源的经济价值。多媒体资源的一部分应用中，要求用户能从加密的资源中得到一部分信息，可以进行浏览，但是未解密时得到的图像质量会比较差，比如付费电视节目、远程教育等。另外，由于在传输过程中可能要经过不同的网络，还要求视频在传输中能适应不同的传输协议和带宽。

传统视频加密后，或者导致视频不可辨识，或者改变原始视频的统计特性，使其难以压缩，或者是加密后的视频时空域的相关性易被利用，从而难以抵抗攻击。

视频加密主要可分为全部加密和部分加密两大类。全部加密的主要缺点是数据量大，不符合实时的要求；而且会产生 marker 和 header emulation，而 marker、header 在压缩位流中是用于同步的。这样，当加密内容在专为未加密压缩位流设计的协议中传输时就会产生问题。针对这些应用产生了选择加密，也称部分加密。关键是选择恰当的部分进行加密。针对不同的视频结构，加密选择的位置不同，主要结合 MPEG 对帧或块加密。按加密的先后可以分为压缩前加密、压缩后加密和同时进行压缩加密三种方法。为了适应各种传输网络，针对第 2 个应用要求，有结合变换编码的加密、结合自适应压缩的加密和其他在传输过程中加密的技术。但不管是哪一种加密，都必须考虑实时性、误码率、压缩率、实现复杂度与现有标准的兼容性问题。

2.8.2 DRM 加密的结构

选择性加密算法中，如果采用 DES，简单地减少一半的加密数据，可能会符合低速率传输的实时要求，但不能满足高分辨率、高信息速率的实时要求，因此有必要对加密的位置进

行分析。在传统的算法里,有只加密头文件的,也有只加密 I 帧的。只加密 I 帧的时间与全部加密的时间相比,减少了 80%~85%,但不够安全,因为 P、B 帧中,高能量的 I、P 块会泄露部分信息。进行了改进后,增加了 I 帧在视频序列的出现频率,可以增加安全性,但是这样做会降低压缩算法的效率,增加计算复杂度,安全性也不够理想。也有人提出对 I 帧的头文件和其他关键信息进行加密,但这样会带来延迟或额外开销。也可以考虑在熵编码前对 DCT 系数进行随机的频域置换,但置换破坏了变换域的能量集中特性,会降低压缩效率,特别是对于低速率视频而言。若是在每个加密的 MB 中控制,使最多只有 64 个运动向量或 DCT 符号位被加密(包括 P、B 帧 MV 的 64 个符号位,非 0 的 DC,最低频的非 0 的 AC 系数),会限定需加密数据的上限,但没有设定下限,安全性不足,特别是对低速或无线的视频。以上算法都是基于不同领域的相对重要性而设计的。

1. 基于 DES 的视频加密

对视频的结构进行分析可以发现:没有加密的 P 帧、B 帧有很多 Intra-coded 宏块(I-MB)。运动补偿区域含有视频的运动信息,这些信息和 I-MB 有一定的比例关系。而 P、B 帧有 25% 的宏块是被编码成 I-MB 的。在有场景变换时,P、B 帧有 100% 的宏块被编码成 I-MB。若不对这些宏块加密,就是对很多信息没有加密。基于以上分析,文献[35]提出了两种视频加密算法。一种是对所有 I-MB(包括 P、B 帧中的 I 块)加密;另一种是对所有 I-MB 加密,并且对 P、B 帧中的非 I-MB 的 header 加密。其中的加密算法是 DES。对 P、B 帧非 I-MB 的宏块的 header 加密有两种方法可用,两种方法各有优缺点。一种符合 MPEG-1 标准,且系数都是 DC 系数或低阶 DCT 系数,含有块中的大多数信息。与对 P、B 帧的 I-MB 加密相似,只是头文件的大小从 27 位到 121 位不等,且分块有所不同。可能的问题是:无法抵抗跳过加密的头文件直接对 DCT 系数解码的攻击。另一种更有效但要传输额外信息。在同一时间片的所有宏块的头文件组合在 sub-bit-stream,称为 header sub-bit-stream。而相应的数据组合成 data sub-bit-stream。把 header sub-bit-stream 以 64 位分组进行 DES 加密,然后把 data sub-bit-stream 与其连接后进行发送。这种方法会产生不符合 MPEG-1 标准的位流。为了正确解码必须传送有关 header sub-bit-stream 的长度的信息。在这些算法中 DES 密钥经常更新并以加密方式传送到接收端。攻击者即使得到其中一个密钥也只能对一部分帧解密。密钥在 MPEG-1 视频位流中传输,可能带来少量的传输开销。DES 密钥在嵌入位流前进行加密。

用 7 个 MPEG-1 视频序列做实验发现,只对 I 帧加密时,图像仍含有原始视频的大量信息。在场景变换的第一个帧中运动补偿失败。播放解码后的视频会清楚地看到视频中物体的运动。对所有 I-MB 加密后,在宏块水平上图像完全被扰乱。跳过加密了的部分而得到的图像会显示更多原始信息,但细节不易被辨认。每个场景开始时的第一帧的运动补偿成功。但是播放解码后的视频仍可清楚看到物体的运动。与前者相比有更好的安全性。若加密所有 I-MB 和所有非 I-MB 的 header,则图像不含任何原始视频的信息。播放解码后的视频也不会看到运动信息,但是操作时间较长。

2. 压缩前进行块加密的框架

文献[36]提出了在压缩前加密,特别指出对运动向量加密的重要性。作者分析,若在压缩前加密,则在中介路由器上易于实现变换编码,因为解压与重新压缩不需要密钥(error-drift-free

变换编码除外)。此外,也易于实现选择加密,因为在频域更容易找出关键的数据,从而实现不同层次的安全性和透明度。也易于找出哪些数据不可压缩。如果在压缩域加密,由于压缩域经常使用可变长编码,在实现选择性加密时会产生额外的处理开销和位溢出,且易受信道错误影响。因为一个 64 位的块是一个整体,块里出现的一个错误会使同步信息错误。而同步信息是藏在加密视频中的,这样,要在网络中恢复错误会比较困难。在频域进行空域加密就不会影响抗误码性,甚至更能抵抗丢包,因为空域上某一部分的图像信息在频域是分布在不同的块中的。而且进行变换域的信号处理,如水印,计算一些统计特性(全局直方图、系数能量、motion intensity 等)就可以不需要解密。另外,其最终的输出会符合压缩标准。若是变换后再加密则易于实现,且增加的处理开销可以忽略,也易于控制透明度。基于以上分析,文献[36]提出了先在变换域加密,再进行压缩。具体步骤是先对输入视频进行频域变换,再把变换系数按块划分,接着进行选择加密(如有选择地对位加密,随机地改变符号,或者置乱块,进行块旋转等)。同时也对运动向量随机改变符号和置乱。整个加密过程用一个密钥控制,最后进行压缩。

算法框图如图 2-26 所示。

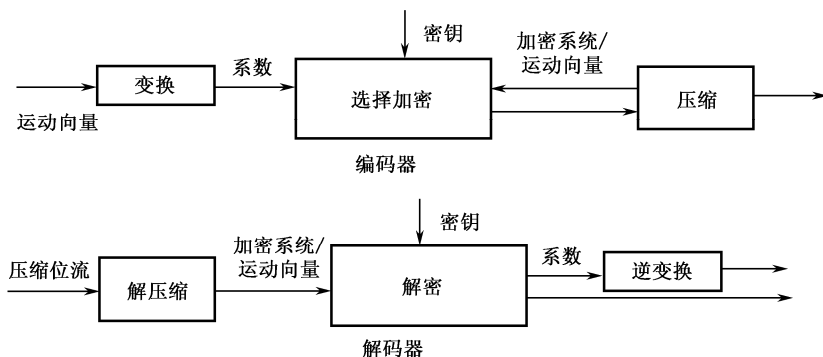


图 2-26 压缩前加密系统框图

这种算法易于实现、安全,且可以提供不同层次的透明度,同时对压缩效率和抗误码性影响较少,进行变换编码或者缩放时既不需要预先解密,也不需要事后再加密。图 2-26 中的加密、解密操作会保留图像的特性,从而不影响压缩算法。虽然框图中两者分开,但加密算法的设计会将压缩算法考虑进去。

文献[36]举了两个具体实现的例子:小波和 DCT 变换后加密。小波变换后加密具体步骤是:先对视频进行垂直和水平的滤波后,再对每帧进行 5 次小波变换,产生 16 个子带。然后在不明显破坏系数的统计特性的前提下进行加密或置乱。可以采取的加密方法有两种:一是加密有较大的熵的数据,二是置乱小波系数。置乱小波系数也可以有三种选择:一是加密某些不能高度压缩的位,这样不会影响编码效率。二是置乱块,先把子带分成相同大小的块,其中不同子带的分块大小可以不相同。用一个密钥生成的置乱表置乱这些块,块里会保存子带的多数二维统计特性,只是块的边界可能会受到一些影响,如果分块较大则安全性较低,但是对编码影响较少。三是块旋转,即把块进行 8 种不同的旋转,从旋转后的 8 个版本中选择一块加密。选择的过程用密钥控制。这三种方法可以分别或综合使用。DCT 变换后加密是对加密 DCT 系数,即先把视频帧分块后,加密 I 宏块,同时也要加密运动向量。试验结果显示小波变换后加密可以使视频有不同程度的透明度,不同的加密方法会带来不同的安全性,

PSNR 值减少较少。DCT 变换后加密中，不同分块会带来不同的安全性和抗误码性，PSNR 值减少也较少。在 DCT 变换加密后，若不对运动向量加密，则从加密了的视频仍然可以看到物体的运动，只是细节难于辨认。在一些应用中，物体的运动是否可以辨认很重要。这也是一种加密 P/B 帧的有效方法。因为重建 P/B 帧依赖于运动向量的准确性。试验结果显示这样可以达到安全性和编码效率的平衡。使用不同方法会带来不同的复杂度。两种方法对速度、安全性、文件大小和透明度等有较好的平衡，适合网络视频应用。

图 2-26 中的加密系统与压缩和传输系统独立，不需要密钥，可进行变换编码和其他操作。对此可以进行进一步的改进。例如，对运动向量的压缩常常是无损压缩，可以利用这一特点再减少加密对压缩效率的影响。在一些应用中，若不用考虑变换编码的问题，可以进一步把加密与压缩整合，以达到更高的压缩效率。

3. 自适应编码与渐进加密相结合的系统框架

文献[37]提出了结合自适应编码与渐进加密技术，支持同时进行变换编码的系统框架。渐进加密就是先把视频分块，再逐步加密不同的块，用已经加密的块加密准备加密的块，解密时也是串行的。安全的可伸缩流（Secure Scalable Streaming, SSS）编码器和译码器如图 2-27 所示，块的大小可以根据应用而定。由于自适应编码中，较早进行编码的单元可以单独解码，不需要较后编码的单元；而渐进加密中，较早加密的块也可以单独解密，不需要后续的加密块，因此可以考虑将自适应编码与渐进加密结合。基本步骤是先把视频分块；再进行自适应编码，形成含相关信息的头数据；然后用渐进加密对除头数据之外的数据加密；最后把头数据与加密后的数据打包。在传输时，为了适应不同的网络，可以根据未加密的头部丢弃部分数据。

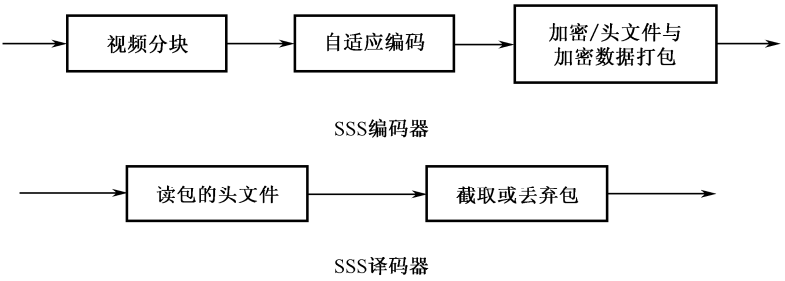


图 2-27 渐进加密技术

算法可以在确保安全性的同时适应不同的传输速率，传输过程不用解密；整个帧编码到同一个位流，有利于安全的率失真优化；传输设备不用保存视频的状态信息，实现简单。有关实例有 JPEG2000、EBCOT、3D 子带编码、MPEG-4 的 FGS。

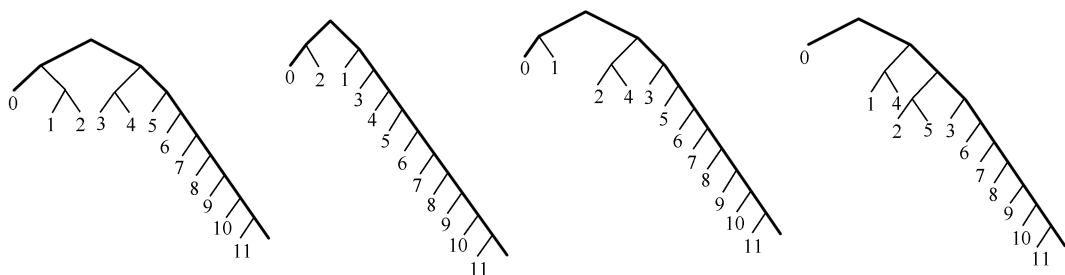
2.8.3 DRM 加密算法

DRM 加密技术的研究中，还有一部分集中在改进加密算法，使加密更快速或更能与现有工具、格式、协议兼容。

1. 结合熵编码器，用统计模型形成密钥

传统的工作主要考虑如何用高阶模型进行自适应算术编码，即在隐藏原始统计模型的同时把它转换成密码。优点是可以利用统计模型多变的特点，且模型空间很大；加密操作不会增加计算量。至于安全性方面，这种方法对唯密文和已知明文攻击稳健，对选择明文攻击比较脆弱，但是在应用中这种攻击不是一种威胁。同时这种方法在文本压缩和加密时非常有效。在多媒体压缩中，Huffman 和 QM 编码器很普遍，且有相似的统计模型。对 Huffman 编码器，统计模型一般是固定大小、非自适应的二值树，而 QM 编码器的初始状态由 3 个整数组成。隐藏 Huffman 编码表不是理想的加密方法。隐藏 QM 编码器的初始态也不能提供安全性，因为尝试所有可能的整数值破解它太容易了。为了解决在一个简单的熵编码器中密钥/模型空间有限的问题，使用 m 个统计模型而不是一个。交替地用 m 个模型对输入的符号流进行编码，当模型以固定、已知的顺序替换，密钥空间只以斜率 m 线性增加。但如果模型的替换顺序是保密的，密钥空间会以 m 的 p 次方增加。其中 p 是隐藏的替换序列的长度。 m 可以很小，因为它与存储统计模型所需的存储空间有线性关系。把 m 设为 4， p 设为 64 就可产生安全、足够大的密钥空间。

文献[38]介绍了有多个 Huffman 编码表的 MHT (Multiple Huffman Tables) 加密算法。多数多媒体压缩系统中的 Huffman 编码使用预定义的 Huffman 表。对每个符号的编码是简单的查表过程，这种方法的速度很快。MHT 加密算法也是对输入数据流用 Huffman 编码表编码，只是把编目表的内容和顺序作为密钥。基本算法包括三步：生成多个不同的编码表，生成随机向量，对不同的符号用不同的编码表编码。还可以做进一步改进使生成的编码表不影响压缩率：用不同的训练图像（或音频）集生成编码表。这样生成的编码表的效果一样，因为每个训练集都是所有音频或图像的对称的表达。万一有两个编码表一样，只要把其中一个挑出来就可以了。更简便的方法是只训练和产生 4 个不同的 Huffman 编码表，再使用 Huffman 树变异算法就可以产生许多不同的编码表。要生成新的 Huffman 树，先随机地生成一个 $(m-1)$ 位的整数，然后把值为 0 的位排列成标准 Huffman 树中相应的对。这种变异不影响编码效率，如图 2-28 所示。



(a) JPEG DC 编码的原始 Huffman 编码树

(b) 分别用三幅图像训练后的 Huffman 编码树

图 2-28 Huffman 编码树

基本 MHT 加密算法每次加密的时间少于一个 CPU 周期。当一个符号用一般的 Huffman 编码器编码，一般把转换量加在表的基本地址之上以获得所需的 Huffman 码的地址。文献[38]的操作与一般编码器不同的是，只需要进行一次载入内存的操作、一次加法和一次比较操作（图 2-29）。

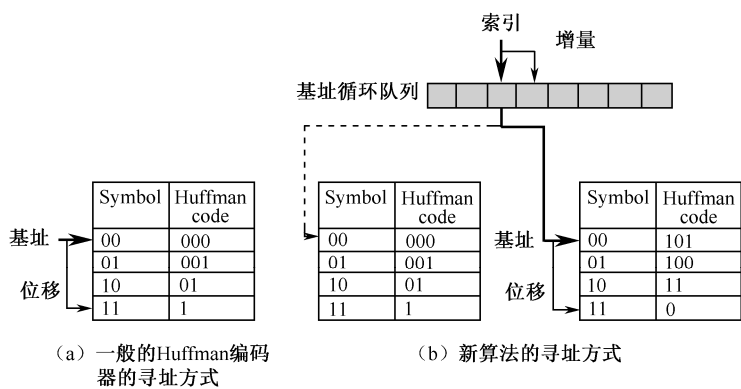


图 2-29 寻址方式

由于有大的密钥空间，对纯密文穷举密钥检索攻击稳健；对已知明文攻击稳健，因为在明文和密文间的同步很难被攻击者找到；但是，如果攻击者可以加密大量的选择明文块并与得到的密文比较，则会对已知明文攻击脆弱。

还可以从以下几个方面进一步改进算法，以降低计算成本：一是随机选择位插入。为了增加寻找同步的难度，可以根据一个密钥把随机位插入加密位流。具体步骤是，先产生一个随机向量，再实现一个函数。这种插入轻微增加了密文的长度。调整参数可以限制密文改变的长度。由于位插入只是每 w 个位插入一次，可减少加密时间。另一种方法是把明文分成几个部分，不同部分使用不同的密钥。这种方法中，将流密码整合进来。一个标准的流密码使用密钥流产生器来产生伪随机二值序列，这个二值序列与明文长度相同。逐位异或密钥流和明文来产生密文，如图 2-30 (a) 所示。为了保证安全性，要确保同样的密钥流不会被使用两次。流密码的计算量主要在密钥流产生器上。一般使用分组密码，如 DES，来实现密钥流产生器。这样流密码的计算成本与分组密码的几乎一样。为了降低密钥流产生器的计算量，把密钥流和明文分别分成 x 位和 y 位的分组。每个明文分组用 MHT 加密算法加密，相应的密钥流分组作为部分密钥，如图 2-30 (b) 所示。这种算法主要解决选择明文攻击问题。在改进的算法中，攻击者所得到的部分密钥并不能解码后续的部分。

另外，文献[38]介绍了使用 QM 编码器的 MSI (Multiple State Indices, 多状态指数) 算法。传统的 QM 编码器的初始概率估计中，状态索引初始为 0，即 0 和 1 是等概率的。由于只有 113 个可能的索引值，把它初始为一个秘密值并不能提供安全性。文献[38]提出了使用 4 个索引，设为隐藏的初始值，根据秘密的顺序交替地使用它们来编码输入位流。具体步骤如下：

- ① 产生随机密钥；
- ② 初始 4 个状态索引；
- ③ 对不同的位用不同的索引去进行概率估计；
- ④ 在需要时进行状态更新。使 4 个索引被同步成相同值。即使状态索引在某些点的值相同，在其它点它们还会改变，而不会一直取相同值。

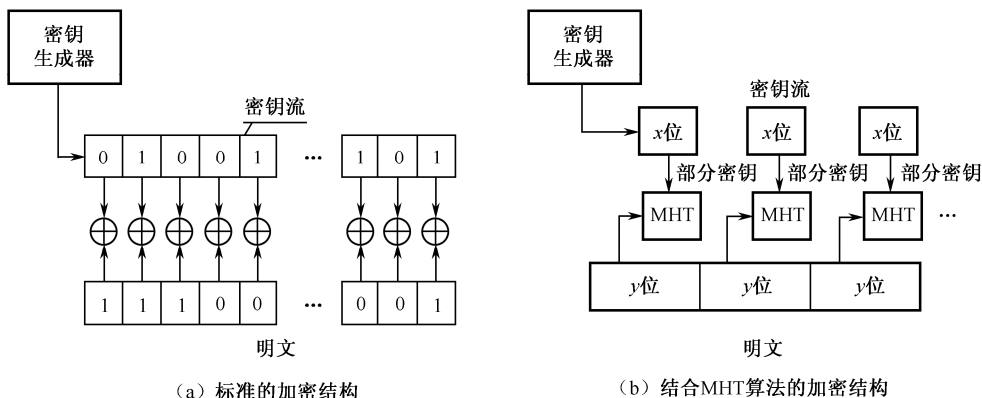


图 2-30 加密结构

主要的计算成本来自第3步和第4步。但是 MSI 算法的计算成本还需要进一步减少。实验结果显示^{[12][38]}，利用 Huffman 编码器加密时，对某些图像，原始 Huffman 树的压缩率会更高。但对另一些图像，MHT 算法效果更好。总体压缩效果比较，MHT 大致与原始 Huffman 编码相同。由于每个符号有不止一个可能的码长，对攻击者而言，比较难以实现同步。每加密一个码的网络计算成本少于一个 CPU 周期。而利用 QM 编码器加密时，每加密一个码需要 6 个 CPU 周期，编码后的数据大小只比原始 QM 编码器大 0.82%~4.16%。

2. 预置的共享的加密算法

在文献[39]中，密钥管理者给传播树的不同节点分配唯一的密钥集。对于给定的节点，激活的密钥集和分配到该节点的密钥集唯一确定一个多项式。对于一个 n 层的视频，在多播结构中需要 n 组密钥。使用以上算法可以同时生成这 n 个密钥。利用子集互不相交的特性，产生不同的密钥。需要更新密钥的时候，只需要改变密钥集的分配。密钥集的数量作为预置信息被保存，如图 2-31 所示。

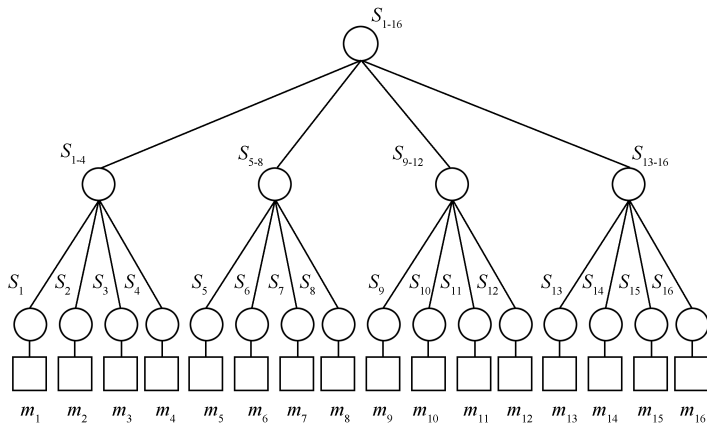


图 2-31 密钥共享的结构

算法的特点是每次加/减用户的时候用于保护新的密钥集分配的密钥都是新的；需要改变

密钥的时候只需改变激活的密钥集；但是由于密钥间的联系较密切，其中一个密钥出错会导致其他相关密钥出错；可能不适用于双重加密协议（Dual-encryption Protocol, DEP），该协议的有些加密部分不需要经常更新密钥；寻找一个节点的密钥的难易度依赖于该节点的多项式，密钥集的数量是预置信息，可根据安全性的要求调整算法。

3. JPEG2000 分层加密

文献[40]算法的主要思想是从一个主密钥生成其他密钥。算法假设原始图像是灰度的，码流只有两种层次。该算法可以扩展到更多层次。

① 加密时生成密钥：把一个主密钥划分成两部分，如图 2-32 所示。

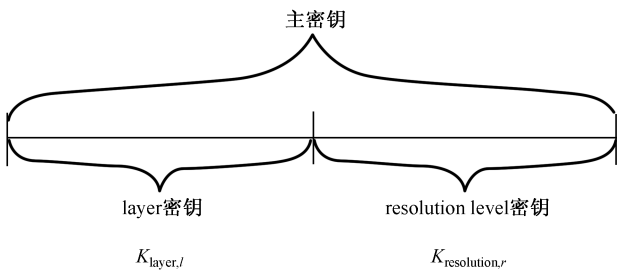


图 2-32 主密钥划分成两部分

② 根据不同层次生成相应的密钥，如图 2-33 所示。

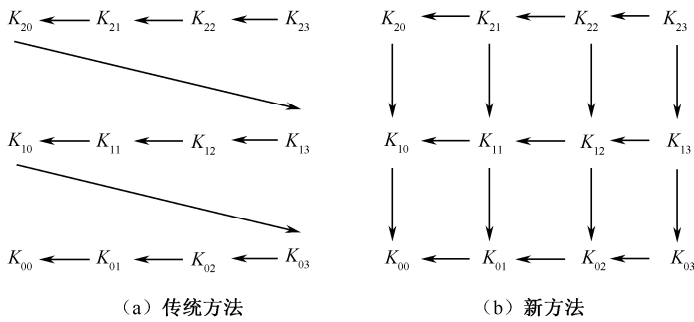


图 2-33 密钥生成的顺序

$K_{layer,L}=H^{l-L}(k_{layer,l})=H(H^{l-L=1}(k_{layer,l}))$ ，其中 $L=l-1, \dots, 2, 1$ ， $K_{resolution,R}=H^{r-R}(K_{resolution,r})$ ，其中 $R=r-1, \dots, 2, 1$ 。

③ 解密时生成密钥。

若分层是 layer，则先假设得到的密钥是 K_{13} ，则由 K_{13} 得到 K_{12} 和 K_{03} 。再由 K_{12} 得到 K_{11} 和 K_{02} ，由 K_{03} 得到 K_{02} 。以此类推到得到 K_{00} 为止，共得到 layer 0、1 的 8 个密钥。若分层是 resolution level，假设得到的是 K_{22} ，则由 K_{22} 得到 K_{21} 和 K_{12} 。再由 K_{21} 得到 K_{20} 和 K_{11} ，由 K_{12} 得到 K_{11} 和 K_{02} 。以此类推直到 K_{00} ，共得到 resolution level 0、1、2 的 9 个密钥。

$K_{layer,L}=H^{x-L}(K_{layer,x})$ ，其中 $L=l-1, \dots, 2, 1$ ；

$K_{resolution,R}=H^{y-R}(K_{resolution,y})$ ，其中 $R=r-1, \dots, 2, 1$ 。

算法可以抵抗合谋攻击。实验图像是 512×512，24 位彩色的。用 JPEG2000 VM8.6 软件

作为编解码器。编码速率是 0.5 位/像素，共有三层（0.1、0.2 和 0.5 位/像素），有 4 个 resolution level。基于 blowfish 用 60 字节的私钥对图像进行加密和解密。每个私钥被平均分成 3 个部分用于三种层次划分（layer，resolution level 和 color component）。

- 对 layer：控制解码后图像的 SNR。
- 对 resolution level：控制解码后图像的大小。
- 对 color component：控制色彩。

可以选择不同的分层方法以提供不同的图像质量。

2.8.4 DRM 加密效果的检验

文献[41]提出了一个模型来衡量选择加密的效果，就是考察用边信息把媒体从扭曲中恢复的复杂度，而不是恢复密钥的复杂度。

1. 密码分析的模型

传统密码分析模型（图 2-34）建模的重点是找到密钥的复杂度，或恢复加密数据的复杂度。但是这些模型未考虑到未加密数据可能会被利用。如果目标不是寻找密钥，已知明文的假设就不起作用。攻击模型可能利用多媒体信息的普遍的统计和结构特性；且由于具有商业媒体的特性，它不能避免被攻击者获得重要信息的情况。新的密码分析模型（图 2-35）更好地衡量商业应用中选择加密的安全性。模型的重点是减少扭曲（增加质量）的复杂度。

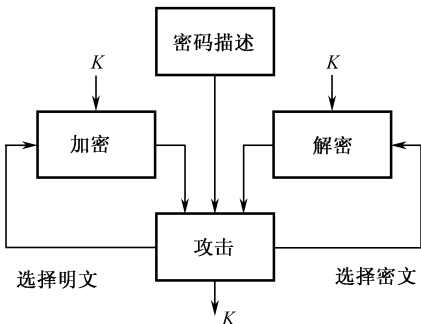


图 2-34 传统密码分析模型

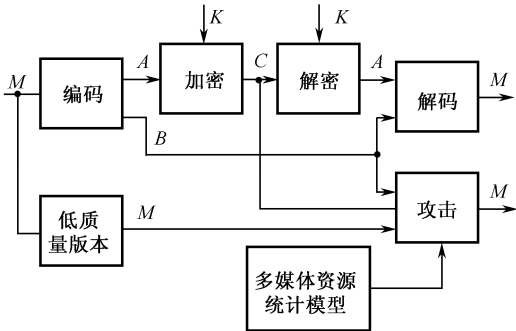


图 2-35 新的密码分析模型

图 2-36 显示了在假设的加密和攻击方法下，复杂度和扭曲的关系。从图中可看出，理想

的情况下，除非用穷举法恢复密钥，否则不能减少扭曲。稍差一点的系统是在付出很大的代价的时候可以减少扭曲。而脆弱的系统是低复杂度的攻击就会很快减少扭曲。因此，即使是脆弱的系统（很容易减少扭曲的系统），要恢复密钥也可能比较复杂。用恢复密钥的复杂度来衡量系统的安全性是不够恰当的。

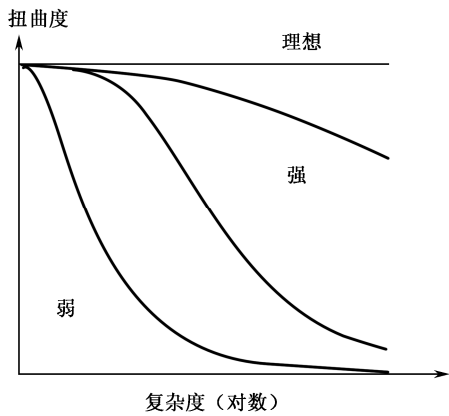


图 2-36 不同的部分加密算法的扭曲度-复杂度曲线

文献[41]举了两个例子。一个是对 DCT 系数符号位的加密使用 AES 算法，因为该算法恢复密钥的复杂度比较高。但加密后产生的是格式兼容的文件，则除了符号的值未知，其他的信息是已知的。可以用多个通道，从未加密的数据中恢复加密的数据，进行攻击。例如，用 AC 系数猜测 8×8 块的符号。文献[41]利用图像的插值和恢复信息进行了攻击。用线性插值和少数通道，可正确恢复 50%~60%的符号位；更复杂的插值和更多通道可以恢复 65%~70%的符号位。即使一些加密的数据未能恢复，还是可以用相对较低的复杂度得到较好的图像质量。另一个是加密重要数据（SPIHT 图像/视频压缩的重要数据的加密）。假设有低质量的图像版本，用 N 位的向量产生图像来替换加密的位。由于猜测不正确的位会产生随机图像，而正确的位会产生正确的图像，可以用这个作为判断所选向量是否恰当的标准。比较攻击图像（用向量产生的图像）和原图像的差异，差异最小时就是所能得到的最好的攻击图像。实验时假设得到的是缩略图，原图像被加密 2%，从少于 2000 个强力攻击测试中可以得到前 480 个加密位的大多数，再用部分解码后的图像移除水印，只要用足够多的时间就可以移除水印，同时加入高频分量（缩略图中没有的分量）。

参考文献

[1] C. E. Shannon.Communication theory of secrecy system.Bell System Technical Journal, 1949,28(4):656-715.

[2] NBS.Data Encryption Standard. U.S.Department of Commerce, FIPS Publication 46, Washington, D.C., January 1977.National Bureau of Standards.

[3] W. Diffie, M.E. Hellman. New directions in cryptography, IEEE Transactions on Information Theory,IT- 22 ,1976, 644-654.

- [4] R.L Rivest, A. Shamir and L. Adleman. A method for obtaining digital signatures and public cryptosystems. Communication ACM 1978,21:120-126.
- [5] T. ElGamal. A public key cryptosystem and a signature scheme based on the discrete logarithm. IEEE Transactions on Information Theory, 1985, 31(4): 469-472.
- [6] V. S. Miller. Use of elliptic curve in cryptography. Advances in Cryptology -CRYPTO'85 Proceedings. Santa Barbara. California. United States: Springer-Verlag. 1986, 417-426.
- [7] National Institute of Standards and Technology, NIST FIPS PUB186, Digital Signature Standard, U.S. Department of Commerce, 1994.
- [8] C. P. Schnorr. Efficient identification and signature for smart cards. Advances in Cryptology.-CRYPTO'89, LNCS435, Springer-Verlag, Berlin, 1990, 239-252.
- [9] D. Chaum. Blind signature for untraceable payments, In Advances in Cryptology, Proc CRYPTO'82, Lecture Notes in Computer Science, Springer-Verlag, 1983, 199-203.
- [10] D. Chaum, E. Van. Heyst. Group signature. Proceedings of EUROCRYPT'91, Lecture Notes in Computer Science. Springer-Verlag, 1991, 547: 257-265.
- [11] Carlisle Adams Steve Lloyd. 冯登国, 等译. 公开密钥基础设计——概念、标准和实施. 北京: 人民邮电出版社, 2001.
- [12] 何妙谊. DRM 加密技术. 中山大学研究生学刊(自然科学、医学版), 2006, 27(2): 31-41.
- [13] 陈鲁生, 沈世镒. 现代密码学. 北京: 科学出版社, 2002.
- [14] 张文涛. 分组密码的分析与设计. 中国科学院研究生院博士学位论文, 2003.
- [15] H. Feistel. Cryptography and Computer Privacy, Scientific American, 1973.
- [16] William Stallings. 孟庆树, 等译. 密码编码学与网络安全——原理与实践 4 版. 北京: 电子工业出版社, 2010.
- [17] 韦宝典. 高级加密标准 AES 中若干问题的研究. 西安电子科技大学博士学位论文, 2003.
- [18] J. Daemen, L. Knudsen and V. Rijmen. The Block Cipher Square, Fast Software Encryption 1997, Lecture Notes in Computer Science 1997. 1267: 149-165.
- [19] J. Daemen. Cipher and Hash Function Design Strategies Based on Linear and Differential Cryptanalysis. Doctoral dissertation, K.U.Leuven, 1995.
- [20] R. C. Merkle, M.E. Hellman. Hiding Information and Signatures in Trapdoor Knapsacks. IEEE Trans Information Theory, 1978, (24): 525-530.
- [21] A. Shamir. A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem. Proc 23rd IEEE Symposium on Foundations of computer science, 1982, 145-152.
- [22] A. K. Lenstra. Integer factoring. Designs, Codes and Cryptography. 19(2000):101-128.
- [23] P. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS and Other Systems. Proc. of CRYPTOLOGY'96. Berlin, Germany: Springer-Verlag, 1996.
- [24] 汪丽. 基于代数方法的 ElGamal 公钥密码体制的建立. 东北大学硕士学位论文, 2008.
- [25] R Rivest. The MD5 Message-Digest Algorithm. RFC 1321, 1992.
- [26] NIST. Secure hash standard. Federal Information Processing Standards, FIPS-180-1, 1995.

- [27] X. Y. Wang, Y. L. Yin, H. B. Yu .Finding collisions on the Full SHA-1.Advances in Cryptology--Crypto'05, LNCS 3621.2005:17-36.
- [28] C. P. Schnorr. Efficient signature generation for smart cards. Advances in cryptology-crypto'89 proceedings, Springer-Verlag, 1991, 239-252.
- [29] C. Popescu. An efficient ID-based group signature scheme, Studia Univ. Babes-bolyai, Informatica, 2002, Vol. XLVII(2), 29-35.
- [30] J. Camenisch, M. Stadler. Efficient group signature schemes for large groups, In Crypto'97, Springer Vertag, 1997, LNCS 1294, 410-424.
- [31] Internet X.509 Public Key Infrastructures. RFC 2459.1999.
- [32] 张宏. 基于 PKI 身份认证系统的研究与实现, 西北大学硕士学位论文, 2008.
- [33] 那什 著, 张玉清, 陈建奇 等译. 公钥基础设施(PKI)实现和管理电子安全. 北京: 清华大学出版社, 2002.
- [34] M. Bertin. Smart card matches fingerprint data for PKI. Biometric Technology. 2000,1(2-3).
- [35] A. M. Alattar and G. I. Al-Regib. Evaluation of selective encryption techniques for secure transmission of PEG-compressed bitstreams. Proc. IEEE Int. Symp. Circuits and Systems. 1999, 340-343.
- [36] W. Zeng and S. Lei . Efficient frequency domain selective scrambling of digital video. IEEE Trans. Multimedia. Mar. 2003(15): 118-129.
- [37] S. J. Wee and J. G. Apostolopoulos. Secure scalable streaming enabling transcoding without decryption . Proc. Int. Con.f Image Processing. 2001(11): 437-440.
- [38] CP Wu and CCJ Kuo. Efficientmultimedia encryption via entropy codec design. Proc. SPIE SecurityWatermarkingMultimedia Contents III. Jan. 2001(4314):128-138.
- [39] AM Eskicioglu and EJ Delp. An integrated approach to encrypting scalable video. Proc. IEEE Int. Con.f Multimedia Expo. 2002(11): 573-576.
- [40] S. Imaizumi, O. Watanabe, M. Fujiyoshi , et a.l Generalized hierararchical encryption of JPEG 2000 codestreams for access control. ICIP 2005.
- [41] A. Said . Measuring the strength of partial encryption schemes. ICIP 2005.

数字水印技术

加密技术对数字内容的版权保护具有一定的局限性，加密内容一旦被解密，信息就完全变成了明文，无法防止数据的非法复制和鉴别数字内容的知识产权。因此，加密技术只能提供数字内容的安全传输，而无法保护数字内容本身。在这种情况下，急需一种行之有效的手段来对数字内容进行版权保护或对内容的真实性与完整性进行认证，数字水印技术就是在这样的背景下产生的。利用数字水印技术，将版权信息嵌入图像、声音、视频等数字产品中，在需要的时候再提取出来作为版权证明或对内容的真实性与完整性进行认证。在这个过程中，信息的嵌入不会对原始的载体信息造成视觉或听觉上的质量下降，而且所嵌入的信息在受到作品传输过程中所引入的干扰与噪声后仍然能够有效地提取。数字水印可以用来证明创作者对其作品的所有权，并作为鉴定、起诉非法侵权的证据，从而成为知识产权保护和防伪的有效手段。

数字水印技术从 20 世纪 90 年代兴起，Van Schyndel 在 ICIP'94 会议上发表了题为“A Digital Watermark”的论文^[1]，它是第一篇在主要会议上发表的关于数字水印的文章。1996 年 5 月，在英国剑桥牛顿研究所召开了第一届信息隐藏技术国际研讨会，会议的一个主要议题就是数字水印技术。此后各种重要的学术会议及学术期刊上不断出现关于数字水印研究的文章，数字水印成为了国际学术界和企业界的一个热门研究领域和发展方向，逐步得到了人们的广泛关注和高度重视。

图像水印技术是研究最早也相对成熟的数字水印技术。文献[1]提出了两种通过修改灰度图像的最低有效位（Least Significant Bit, LSB）来嵌入水印信号的方法。LSB 方法简单易行，但其鲁棒性较差。随着数字水印技术研究的不断深入，数字水印技术从基于载体信号的空域进行研究发展到基于载体信号变换域进行研究，例如，Cox 等人基于扩频通信的思想，提出了将水印信号嵌入视觉最重要的频域系数上^[2]。但这些水印算法主要针对常规的信号处理攻击，如 JPEG 压缩、噪声攻击、图像增强、图像滤波等。当发生旋转、缩放、平移等几何变换的时候，很难成功地进行水印检测。因此，几何攻击也被认为是水印技术通向应用的瓶颈^[3]。

3.1 数字水印概述

从信号处理的角度看，数字水印相当于在强背景下叠加了一个弱噪声。从数字通信的角度看，数字水印可以理解为在一个宽带信道（载体信号）上用扩频通信技术传输了一个窄带信号（水印）。

3.1.1 数字水印的系统模型

不管是基于空域的还是基于变换域的数字水印技术，一个完整的数字水印系统的设计一般包括三部分：水印生成、水印嵌入和水印检测。

1. 水印生成

水印信号的产生可基于伪随机序列发生器或混沌系统，也可以是有意义的二值、灰度、彩色图像。为了携带更多的版权信息，一般采用二值图像或灰度图像来表示水印，例如产品的序列号、logo 等。对于有意义的水印序列，为了提高水印信息的安全性，增强水印抵抗恶意攻击的能力，可以使用置乱技术对水印进行预处理，以去除水印信息的相关性。有许多有效的置乱方法，如 Aronld 变换、Hilbert 曲线、幻方、广义 G-ray 码等。

2. 水印嵌入

水印嵌入是把水印信息嵌入载体图像中，嵌入过程如图 3-1 所示。

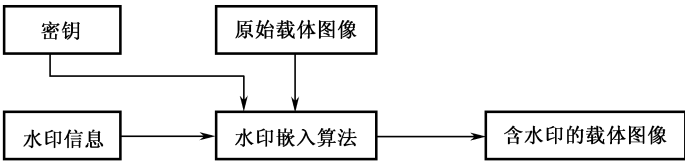


图 3-1 水印嵌入过程

3. 水印检测

水印提取和检测就是对载体图像进行检测，看其中是否含有水印信息，或把水印提取出来。水印提取检测算法要根据水印嵌入算法而采取不同的措施。水印检测过程如图 3-2 所示。

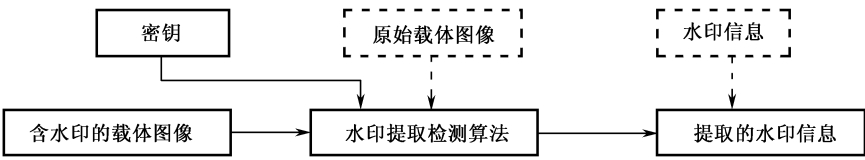


图 3-2 水印检测过程

3.1.2 数字水印的分类

数字水印的分类方法有很多种，分类的出发点不同导致了分类结果的不同，它们之间既有联系又有区别。最常见的分类方法有下列几种。

1. 按数字水印的作用划分

数字水印按其作用的不同可以分为鲁棒水印和脆弱水印。

鲁棒水印主要用于对多媒体产品进行版权保护，即在数字内容中嵌入标识著作权的信息，它要求嵌入的水印能够经受各种常规信号处理操作及攻击。因此，鲁棒水印最主要的特征是水印信息的稳定性。

脆弱水印主要用于完整性保护和多媒体内容认证，根据其作用原理不同又可分为完全脆弱水印和半脆弱水印。完全脆弱水印主要用于多媒体内容的完整性认证，当图像的任何部分被篡改时，都会反映在水印信号上，适用于将待保护的数据看成一个整体，其完整性不可破坏的情形。半脆弱水印对于一般攻击具有鲁棒性，而对于某些恶意的攻击，则可以对篡改进行检测，往往还要对篡改的区域进行定位。由于图像与视频压缩等操作对于多媒体的信息传输是必要的，不会对多媒体内容的质量造成很大的破坏，需要水印算法能抵抗这些攻击。因此，半脆弱水印相比于完全脆弱水印具有更大的应用价值。

2. 按数字水印的检测方法划分

数字水印按其检测方法不同可以分为非盲水印、半盲水印和盲水印。

非盲水印在检测过程中需要原始数据；半盲水印的检测过程不需要原始数据，但需要一些参考信息（一般的自适应水印算法属于这一类型）；而盲水印的检测只需要密钥，不需要原始数据（图 3-2）。

一般而言，非盲水印的抗攻击能力较强，但只能应用在那些可以得到原始图像的场合。例如，在所有权认证过程中，作品的所有者可以拿出原作品进行检测。盲水印技术则更加具有实用性，尤其是面对大数据量且须考虑实时性的视频信息时，此时盲水印的商业价值将得到充分体现。

3. 按数字水印隐藏的位置划分

数字水印按其隐藏的位置不同可以分为空域水印、频域水印和混合水印。

空域水印是直接在原始信号空间上叠加水印信息，如图像水印中直接修改像素最低有效位的 LSB 算法。空域水印实现简单、计算量小。而频域水印是在变换域（如 DCT 域、DWT 域等）上隐藏水印，变换域水印有较强的鲁棒性，且可较好利用人类视觉/听觉特性。混合域水印吸取了空域与变换域水印的优点。随着数字水印技术的发展，各种水印算法层出不穷，水印的隐藏位置也不再局限于上述几种，实际上只要构成一种信号变换，就有可能在其变换空间上隐藏水印。

3.1.3 数字水印的性能分析

数字水印算法有多种评估标准，主要有以下几种客观的评价标准。

1. 鲁棒性

鲁棒性指在经过常规信号处理操作及对水印系统的攻击后仍能够检测出水印的能力。常规信号处理操作主要包括加噪、滤波、压缩等；常见的攻击有几何变形、剪切、旋转篡改水印等。

为了对水印算法的鲁棒性进行客观评价，计算原始水印与提取的水印的相关系数（Normalized Correlation, NC），通过相关系数 NC 的值来比较两者之间的相似度。若 NC 值等于 1 则两者完全一样，接近 1 表示两者相似，对于鲁棒水印要求在信号失真的情况下依然能得到较大的相关系数，而脆弱水印则在可信度受到破坏时得到较小的相关系数。

$$NC = \frac{\sum_{i,j} W(i,j) \times W^*(i,j)}{\sqrt{\sum_{i,j} W^2(i,j) \times W^{*2}(i,j)}} \tag{3-1}$$

式中，W 为原始水印，W* 为提取的水印。

2. 不可感知性

不可感知性指在数字作品中嵌入数字水印不会引起明显的降质和视觉效果的明显变化。为了定量地确定不可感知性，把嵌入的水印信号看成加载到载体图像上的噪声，使用峰值信噪比（Peak Signal-to-Noise Ratio, PSNR）来评估嵌入水印后图像质量的改变状况。

$$PSNR = 10 \lg \left(\frac{\max(I^2(i,j))}{E} \right) \tag{3-2}$$

$$E = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (I(i,j) - I^*(i,j))^2 \tag{3-3}$$

式中，M×N 是图像的大小，I(i,j) 与 I*(i,j) 为嵌入水印前后的图像的灰度值，E 为 I(i,j) 与 I*(i,j) 的均方差误差。PSNR 的单位是分贝（dB）。

PSNR 用来评估原始图像和含水印图像之间的相似程度，一般在 35dB 左右就意味着水印几乎是不可感知的。但是这只是一个经验的测量值，有些时候并不能正确反映图像质量的改变程度。

3. 水印容量

图像水印容量是指在载体图像中可以隐藏的最大水印信息量。水印容量取决于载体图像的统计特性，失真限度，以及水印嵌入和提取算法是否能充分利用载体图像。对于不同的应用场合，对水印容量的要求各不相同。

水印的鲁棒性、不可感知性和容量之间的关系可以用图 3-3 表示。当三者中的任意一个量固定时，剩下的两个量是相互矛盾的。例如当水印容量一定的情况下，为了获得更好的鲁棒性可以提高水印嵌入的强度，而嵌入强度的提高必然会带来更大的失真。类似地，较低的

水印强度虽可以保证较好的图像质量，但是水印的鲁棒性必然较差。因此，在水印算法设计的过程中往往需要根据实际应用的需要在水印鲁棒性、不可感知性和容量之间寻求折中^[14]。

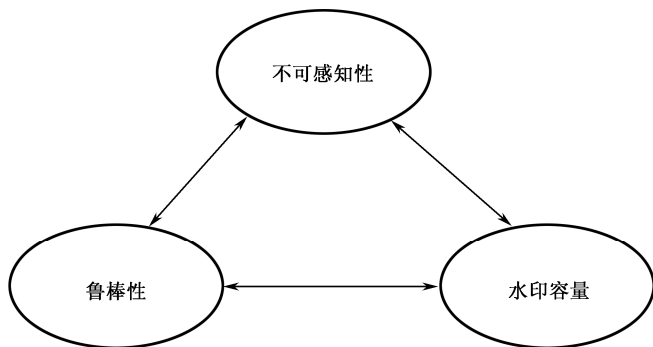


图 3-3 水印性能指标之间的关系

3.2 空域图像水印算法

空域图像水印算法的研究比较早，早期的数字水印算法从本质上来说都是属于空域的。空域图像水印是在图像的空域中嵌入水印的技术，即直接在信号空间上叠加水印信息。空域图像水印的典型算法有 LSB 算法和 Patchwork 算法等。

1994 年，Van Schyndel 在 ICIP'94 会议上发表了题为“A digital watermark”的论文，这是一篇具有历史价值的文献，文中阐明了一些关于水印的重要概念和鲁棒水印检测的通用方法（相关性检测方法），提出了 LSB 图像水印算法。此算法使用特定的密钥通过一个 m 序列（Maximum Length Raildom Sequence）发生器产生随机序列信号，然后按一定的规则将其重新排列成二维水印信号，并按像素点逐一插入原始图像对应像素的最低比特位。

LSB 算法利用了图像的视觉冗余，本质上是一种修改替换方法。对于灰度图像，人眼不能分辨全部 256 个灰度等级，4 个左右灰度等级的差异人眼是不能区别的，而当对比度比较小时，人眼的分辨能力更差。图像像素点的最低几位数据代表的能量很少，更改对灰度的影响很少，称为最低有效位，也称最不显著位，可利用它来隐藏信息。

考虑以一幅 256 色（8 位）灰度 Lena 图像，我们先看看 8 位数据的各数据位对图像的影响。图 3-4 是将 Lena 图像各像素位分别提取出来并转换成二值图像所得到的结果。

可以看到，数据的最低两位看起来像噪声，在视觉上与原图像没有相关性；从低位第 3 位才能看出与原图有较少的联系；第 5、6、7 位包含了图像的大部分信息。

将 Lena 图像各像素最后几位数据分别经过随机化之后得到的结果如图 3-5 所示。

可以看到如果改变每个像素 8 位中的最后两位甚至三位，人眼几乎分辨不出有任何区别。当从改变最后四位开始，人眼开始感觉到区别，并随着随机位数的增加，图像与原图像的区别愈加明显。

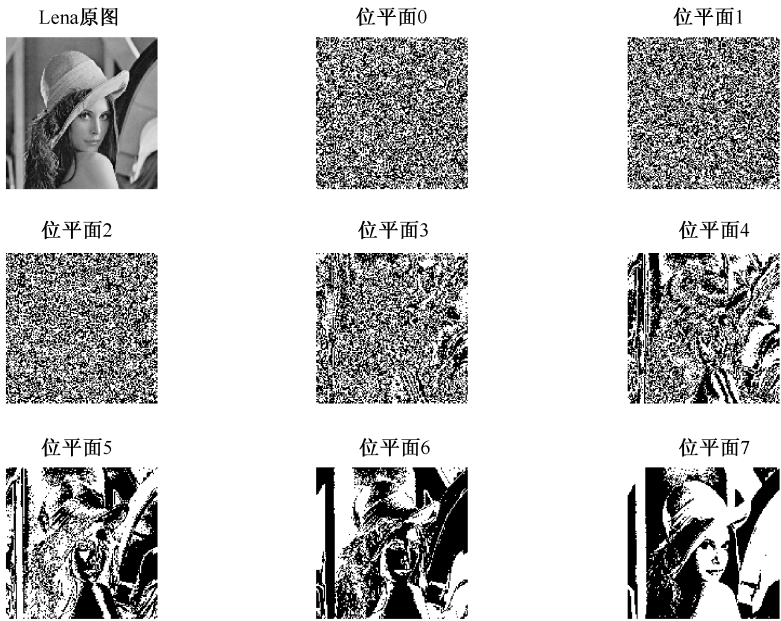


图 3-4 Lena图像各像素位分解

从各位对像素值的影响的分析看，最低一位有的改变对原数值的影响仅为 1，第二位的影响为 2，每三位为 4，每四位为 8，依次类推。位越高对原数值的影响越大，即可以利用低位对视觉的低相关性，将嵌入对象的数据存放到最低的几位中来隐藏信息。

利用最低有效位隐藏信息，可嵌入的信息量根据使用的最低有效位的位数而定。可嵌入的信息量是指载体图像中可容纳的信息量。如果使用最低一位，则可嵌入的信息量是原图像信息的 1/8，如果是最后两位，则是 1/4。使用的位数越小，则嵌入信息对原图像的影响就越小。因此在原图像足够大的情况下，使用的位数越少越好。

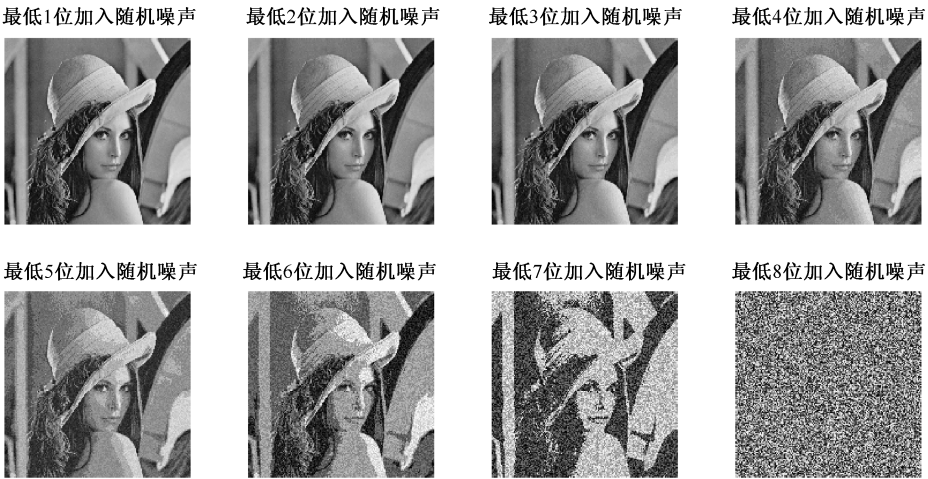


图 3-5 Lena图像各像素位加入随机噪声

令原图像点阵图大小为：宽 W 像素，高 H 像素，颜色分量数为 N ，每一颜色分量用 B 位，使用 b 位用于嵌入信息，则图像可嵌入的信息量 M 为

$$M = \frac{W \times H \times b}{8} (b < B) \quad (3-4)$$

这种算法的优点是应用简单，成本低，可容纳的信息量大，适合于对水印信息隐藏要求不高的场合。由于 LSB 位平面携带着水印，因此在嵌入水印图像没有产生失真的情况下，水印的提取很简单，只需要提取含水印图像的 LSB 位平面即可，而且这种方法是盲水印方法。

但是 LSB 的最大缺陷是对信号处理和恶意攻击的稳健性很差，鲁棒性较弱。对含水印图像采用 StirMark 模拟多种攻击，测试结果表明，使用 LSB 方式嵌入的水印信息的图像在被攻击后，如对图像进行滤波、图像量化、几何变形等处理后，水印信息很容易被破坏，基本上检测不到水印信息。

Patchwork 算法^[5]是一种基于图像数据统计特性的空域水印算法。“Patchwork”一词原指一种用各种颜色和形状的碎布片拼接而成的布料，它形象地说明了该算法的核心思想，即在图像域上通过大量的模式冗余来实现鲁棒数字水印。该算法首先随机选取 N 对像素点 (a_i, b_i) ，然后将每个 a_i 点的亮度值加 1，每个 b_i 点的亮度值减 1，通过增大像素对中一个点的亮度值而相应减小另一个点的亮度值来隐藏信息，这样可以保持整个图像的平均亮度不变。为增强其水印的鲁棒性，还可以把像素对扩展为小块的像素区域（如 8×8 ），通过增大一个区域中的所有像素点的亮度值而相应减小对应区域中所有像素点的亮度值来隐藏信息。适当地调整参数，Patchwork 方法对 JPEG 压缩、FIR 滤波以及图像裁剪具有一定的抵抗力。但该算法水印嵌入容量低，且对串谋攻击抵抗力弱。

3.3 DCT 域水印算法

离散余弦变换是信号处理中常见的一种时域频域变换。与空域图像水印相比，DCT 域的图像水印具有更强的稳健性，同时又与常用的图像压缩标准 JPEG 兼容，二维 DCT 是目前最常用的有损数字图像压缩系统——JPEG 的“核心”，因此 DCT 域的数字水印受到了广泛的重视，是目前研究最多、使用最广泛的数字水印技术。

3.3.1 离散余弦变换的基本概念

傅里叶变换需要计算的是复数，不是实数，而复数运算要比实数运算费时得多。如果采用其他合适的完备正交函数系来代替傅里叶变换所利用的正、余弦函数构成的完备正交函数系，就可以避免这种复数运算。离散余弦变换（Discrete Cosine Transform, DCT）就是基于实数的正交变换。DCT 变换是 N.Ahmed、T.Natarajan 和 K.R.Rao 在 1974 年提出的。对于 $M \times N$ 的像素块，下面给出其二维 DCT 变换的定义。

☒ 定义 3-1 二维离散余弦变换

对于一个 $M \times N$ 二维离散信号 f ，其 DCT 变换为

$$F(u, v) = c(u)c(v) \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i, j) \cos\left(\frac{(2i+1)u\pi}{2M}\right) \cos\left(\frac{(2j+1)v\pi}{2N}\right) \quad (3-5)$$

$$u=0, 1, 2, \dots, M-1$$

$$v=0, 1, 2, \dots, N-1$$

其逆变换为

$$f(i, j) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} c(u)c(v) F(u, v) \cos\left(\frac{(2i+1)u\pi}{2M}\right) \cos\left(\frac{(2j+1)v\pi}{2N}\right) \quad (3-6)$$

$$i=0, 1, 2, \dots, M-1$$

$$j=0, 1, 2, \dots, N-1$$

其中

$$c(u) = \begin{cases} \frac{1}{\sqrt{M}}, & u = 0 \\ \sqrt{\frac{2}{M}}, & u = 1, 2, \dots, M-1 \end{cases} \quad (3-7)$$

$$c(v) = \begin{cases} \frac{1}{\sqrt{N}}, & v = 0 \\ \sqrt{\frac{2}{N}}, & v = 1, 2, \dots, N-1 \end{cases} \quad (3-8)$$

3.3.2 基于 DCT 变换的水印嵌入和提取算法

典型的 DCT 域算法是由 Cox 等人提出的一种基于 DCT 变换的扩频水印技术^[12]。它将满足正态分布的伪随机序列加入图像的 DCT 变换后视觉最重要系数中，它利用了序列扩频技术和人类视觉特性（Human Visual System, HVS）。算法原理为先选定视觉重要系数，再进行修改，利用加法或乘法准则完成水印的嵌入：

$$v'_i = v_i + \alpha w_i \quad (\text{加法准则}) \quad (3-9)$$

$$v'_i = v_i + (1 + \alpha w_i) \quad (\text{乘法准则}) \quad (3-10)$$

其中， v_i 、 v'_i 分别是修改前和修改后的频域系数， α 是缩放因子， w_i 是水印第 i 位。

一般来说，乘法准则的抗失真性能要优于加法准则。水印的检测是通过计算相关函数实现的。从嵌入水印的图像中提取是嵌入规则的逆过程，把提取出来的水印与原水印作相似性运算，与指定的阈值比较，可确定是否存在水印。

Cox 算法虽然没有区分对待各 DCT 系数上的嵌入强度，但它却是鲁棒水印的奠基性算法，得到了非常广泛的应用。HVS 的出现，也进一步推动了扩频水印技术的应用。随后许多人在此基础上进行了大量的工作，结合基于分块 DCT 的图像压缩方法，将水印嵌入受攻击影响较小的系数中，结合 HVS 特性，设计图像自适应算法。例如，Tao 等人提出了一种自适应 DCT 水印技术^[6]，通过利用 HVS 的掩蔽效应，用不同的方法来确定噪声的灵敏度，从而实现一种嵌入强度自适应的水印算法。Podilchuk 等人提出了一种可感知水印的算法，用临界可见误差

来确定水印的最大嵌入能量,使用 HVS 确定在图像的各个部分所能容忍的水印信号的最大强度,从而能很好地实现不可见性和鲁棒性的折中^[7]。Barni 等人提出一种利用 HVS 掩蔽特性的基于 DCT 的水印算法^[8],将水印信息隐藏在 DCT 域的固定中频段,而且在没有原始图像时仍能确定水印隐藏的位置,从而将 Cox 的算法发展成盲水印算法。

Barni 算法在水印嵌入阶段,对 $N \times N$ 的图像进行 DCT 变换,并对 DCT 系数按 Zig-Zag 扫描重新排列为一维向量,对第 L 个系数后面的 M 个系数进行修改以嵌入水印。设开始的 $L+M$ 个 DCT 系数是 T :

$$T = \{t_1, t_2, \dots, t_L, t_{L+1}, \dots, t_{L+M}\} \quad (3-11)$$

水印由 M 个符合正态分布的实数随机数组成:

$$X = \{x_1, x_2, \dots, x_M\} \quad (3-12)$$

按下式将水印嵌入 T 中:

$$t'_{L+i} = t_{L+i} + \alpha |t_{L+i}| x_i, \quad i=1, 2, \dots, M \quad (3-13)$$

改变不同的 DCT 系数对水印系统的整体性能有着不同的影响。

一方面,一般的图像处理均发生在图像频谱的高频区域中,例如对图像进行 JPEG 压缩,它保留了图像的低频信号,而将一部分的高频信号滤掉。如果水印嵌入在高频区域,虽然其不可感知性效果很好,但水印经过图像处理后容易被删除,其鲁棒性差,因此变换域的水印算法不能将水印信号嵌入图像的高频区域中。

另一方面,图像的主要能量集中在它的低频系数上,低频系数直接影响到人眼的视觉效果。在低频系数中嵌入水印信号,由于低频系数携带了图像的大部分能量,图像受到攻击时,攻击者一般要保证图像的质量不能下降得太厉害,这样大部分的低频系数仍然保留,嵌入它们的水印信号因此存活下来,这样能够保证水印的鲁棒性。但是在嵌入的过程中要修改低频系数,对图像重要信息会造成破坏,就会降低图片的视觉效果。

所以一般采用折中的办法:选择中频系数,寻求不可感知性与鲁棒性的兼顾。选取 DCT 中频作为水印嵌入区域,问题就简化为如何在水印的不可感知性与鲁棒性平衡的情况下,把水印嵌入图像的中频系数中。

变换域算法有以下几个特点:

- ① 变换域内嵌入水印信号的能量可以分布到整个图像的像素上,有利于提高鲁棒性;
- ② 可以结合 HVS,有利于提高水印的不可见性;
- ③ 变换域算法可以与当今大部分国际图像和视频压缩标准兼容,尤其是 DCT 域图像水印因其与图像压缩标准 JPEG 兼容,所以得到了广泛重视。

3.4 DWT 域水印算法

小波变换不仅可以对信号进行频率分析,而且较好地解决了突变信号和非平稳信号的分析问题,而傅里叶变换对这些信号处理起来较困难。小波变换是傅里叶变换的进一步发展,它可以更有效地分析局部信号,可以对局部信号进行频率变换。

小波变换是图像压缩标准 JPEG 2000 中的一项关键技术,同时 HVS 在同一频带范围内对不同方向的纹理细节信息表现出的灵敏度不同,这正与离散小波变换具有多分辨分析特征相

类似，因此，人们越来越多地将离散小波变换应用到数字水印中。

3.4.1 小波变换的基本概念

傅立叶变换能够用正弦函数之和表示任何分析函数——甚至是一个狭窄的瞬态信号。然而，这是通过错综复杂的安排，以消去一些正弦波（通过相互抵消）的方式，构造出在大部分区间都为零的函数实现的。这对于可逆变换来说是一个有效的方法，但它却使此函数的频谱图呈现一幅相当混乱的构成。

为了克服这些缺陷，数学家和工程师们已经开发出若干种使用有限宽度基函数进行变换的方法。这些基函数不仅在频率上而且在位置上是变化的，它们是有限宽度的波，被称为小波（wavelet）。基于小波的变换被称为小波变换。

与傅里叶变换一样，小波变换的基本思想是将信号展开成一族基函数之加权和，即用一族函数来表示或逼近信号或函数。这一族函数是通过基本函数的平移和伸缩构成的。

☒ 定义 3-2 连续小波变换

设 $x(t)$ 是平方可积函数，记为 $(x(t) \in L^2(R))$ ， $\psi(t)$ 称为基本小波或母小波的函数，则 $x(t)$ 的小波变换定义如下：

$$WT_x(a, b) = \frac{1}{\sqrt{a}} \int x(t) \psi^* \left(\frac{t-b}{a} \right) dt = \langle x(t), \psi_{a,b}(t) \rangle \quad (3-14)$$

其中， a 是尺度因子， $a > 0$ ； b 是位移因子， $b \in R$ ； $\psi^*(\cdot)$ 是 $\psi(\cdot)$ 的复共轭， $\psi(t) \in L^2(R) \cap L^1(R)$ 且 $\int_{-\infty}^{\infty} \psi(t) dt = 0$ 。

为了在计算机上有效地实现小波变换，要将连续的小波变换离散化。对 a 离散化的方法是按幂级数的形式逐步加大 a ，一般为二进制离散，即离散小波变换（Discrete Wavelet Transform, DWT）主要就是建立在二进制小波变换的基础上的。

☒ 定义 3-3 离散小波变换

对 a 离散化令 $a = 2^j, j > 0, j \in Z$ ，对 b 离散化令 $b = KT_s 2^j$ ， T_s 为时间采样间隔。此时小波函数序列可以表示为

$$\psi_{j,k}(t) = 2^{j/2} \psi(2^{-j}t - k) \quad (3-15)$$

任意函数 $x(t)$ 的二进制离散小波变换为

$$WT_x(j, k) = \int x(t) \cdot \psi_{j,k}^*(t) dt \quad (3-16)$$

可以这样理解小波变换的定义：我们用镜头观察目标 $x(t)$ （即待分析信号）， $\psi(t)$ 代表镜头所起的作用（如滤波或卷积）。 t 相当于使镜头相对于目标平行移动， a 的作用相当于镜头向目标推进或远离。故小波变换具有以下特点。

① 多分辨率（multi-resolution），也叫多尺度（multi-scale）的特点，可以由粗及精地逐步观察信号。在低频段可用高频率分辨率和低时间分辨率（宽分析窗口），在高频段可用低频率分辨率和高时间分辨率（窄分析窗口）。

② 小波变换可以看成用基本频率特性为 $\psi(t)$ 的带通滤波在不同尺度下对信号滤波。由傅

里叶变换的尺度特性可知这组滤波器具有品质因数恒定，即相对带宽（带宽与中心频率之比）恒定的特点。

③ 适当地选择小波，使 $\psi(t)$ 在时域上为有限支撑， $\psi(w)$ 在频域上也比较集中，就可以使小波变换在时、频域都具有表征信号局部特征的能力，因此有利于检测信号的瞬态或奇异点。

④ 小波变换实现上有快速算法（Mallat 小波分解算法）。

正是由于上述特性，人们把小波誉为分析信号的数学显微镜。在二维情况下，小波分析除了“显微”能力外，还具有“极化”能力，即方向选择性，因而引人注目。

3.4.2 数字图像的离散小波变换

数字图像是离散的二维信号，对图像进行二维离散小波变换，最常用的方法是考虑二维尺度函数可分离的情况，也就是

$$\phi(x,y)=\phi(x)\phi(y)$$
 (3-17)

其中， $\phi(x)$ 为一维尺度函数。

二维小波变换由以下三个二维基本小波确定：

$$\begin{cases} \psi^{LH}(x,y)=\phi(x)\psi(y) \\ \psi^{HL}(x,y)=\psi(x)\phi(y) \\ \psi^{HH}(x,y)=\psi(x)\psi(y) \end{cases}$$
 (3-18)

用 $\Omega=(LH,HL,HH)$ 表示小波函数的三个方向。在小波变换中，尺度函数 $\phi(x)$ 可以看成低通滤波器，而同一层的高通滤波器即为小波函数 $\psi(x)$ 。一幅 $n\times n$ 的图像 $f_j(x,y)$ ，其中 n 是2的幂， j 表示分辨率参数，尺度是 2^j ，对于 $j=0$ ，原图像的尺度为1，它是第0层分辨率信号。随着 j 的值每增加1，则尺度加倍，分辨率减半。二维离散小波变换按如下方式进行：在变换的每一层，图像都被分解为四个四分之一大小的图像。

对一幅图像来说，小波变换构成了对它的多尺度的时频分解。图 3-6 给出了对 Lena 图像的两个尺度的分解及图像的二层小波分解示意图。左上角(LL_2)是最低频段滤波后的低尺度逼近，同层分辨率下， HL_2 块包含了水平方向高通、垂直方向低通滤波后所保留的细节信息。同样， LH_2 块保留的是水平方向低通、垂直方向高通滤波后所得的细节信息， HH_2 块包含的是水平和垂直方向都经过高通滤波后的细节信息。相同的处理过程在中分辨率和高分辨率层重复进行。

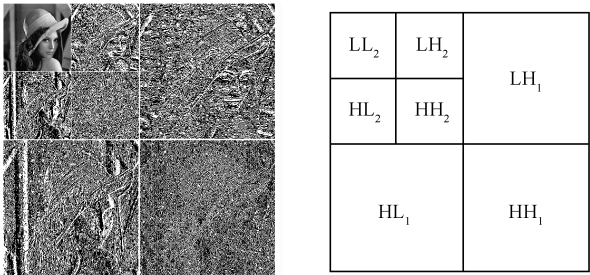


图 3-6 Lena 图像的小波变换

对 256×256 的 Lena 图做三层小波分解，分解后的系数均值、方差和能量统计如表 3-1 所示^[9]。

表 3-1 Lena 图小波分解系数统计表

图 号	最 小 值	最 大 值	均 值	方 差	能 量
LL ₃	145.9	1656.0	781.4	370.4	781.7
HL ₃	-505.3	501.8	-3.1	113.4	11.1
LH ₃	-328.1	379.9	-1.8	61.7	8.1
HH ₃	-214.8	335.1	-0.1	49.4	7.0
HL ₂	-241.7	339.0	-0.9	43.2	6.6
LH ₂	-207.2	234.0	-0.5	26.8	5.2
HH ₂	-122.0	158.7	0.1	18.6	4.3
HL ₁	-129.5	163.0	-0.2	16.6	4.1
LH ₁	-126.0	103.0	-0.1	10.7	3.3
HH ₁	-64.0	78.5	-0.03	7.1	2.6

从图 3-6 和表 3-1 可以看出，经过小波变换后，图像的低频子带携带了图像的大部分信息，因此可以嵌入更多的水印信息，使水印鲁棒性更好，但同时也产生了问题，即图像低频子带的变化容易导致较大的图像失真。相反，高频子带携带的是图像的边缘和纹理信息，人眼对这部分信息不敏感，在高频部分嵌入水印，可以避免引起图像的失真，但同时水印容易遭到破坏。因此，一个有效的水印算法必须在鲁棒性和图像的失真度之间取得平衡。

Haar 小波的性能优良，计算复杂度低于其他小波，而且具有线性相位，可避免小波分解和重构时的图像失真，因此 Haar 小波比较适合于图像水印。在 DWT 域嵌入水印，小波基的选择非常重要，因为选择不同的小波基对嵌入水印的性能有很大影响。

3.4.3 基于 DWT 的水印算法

文献[10]是最早提出小波域数字水印算法的文章之一，该算法把随机扩频序列嵌入小波分解后的左上角的低频部分，在检测时需要用到原图，为非盲水印算法。

D. Kundur 和 D. Hatzinakos 提出了一种按照小波分解层次自适应的数字水印算法^[11]，将图像和需要嵌入的水印信息分别做小波分解，根据 HVS 特性进行数据融合。水印信号是一个二值图像，原图是水印图像大小的 2^m 倍。原图经过 L 层小波变换，水印图像经过一层小波变换，变换后把原图的细节子图分成大小和水印大小相等的不重合的矩形，每个矩形和水印的小波变换矩阵做数字融合，完成水印的嵌入。D. Kundur 等人在嵌入时考虑了 HVS，加入了与局部 HVS 特征相关的水印强度系数，提高了算法的性能。具体嵌入方法如下。

对原图中每个小矩形，计算：

$$C(u,v)=5.05e^{-0.178(u+v)}(e^{0.1(u+v)}-1)$$
 (3-19)

C(u,v)是对比度敏感矩阵，u，v 为空间频率，单位是度。

令 $f_{k,l}^i(m,n)$ 为原图的小波变换矩阵中第 L 层第 k 个细节信息第 i 个分块， $F_{k,l}^i(u,v)$ 为其离散傅里叶变换， $w_{k,l}(m,n)$ 为水印的 DWT 变换矩阵。则每个矩阵对应的强度系数为

$$S(f_{k,l}^i(m,n)) = \sum_{\forall(u,v)} C(u,v) |F_{k,l}^i(u,v)|^2 \quad (3-20)$$

嵌入公式:

$$g_{k,l}^i(m,n) = f_{k,l}^i(m,n) + \gamma_{k,l} \sqrt{S(f_{k,l}^i(m,n))} w_{k,l}(m,n) \quad (3-21)$$

$\gamma_{k,l}$ 是用来平衡不可见性和鲁棒性的系数, 往往与应用有关, 文献[11]中推荐了一个计算方法。

该算法水印检测需要原图, 是非盲水印。该算法虽然是变换域数字水印, 但考虑了图像空域的 HVS 特性, 是比较有影响的数字水印算法。

D. Kundur 和 D. Hatzinakos 在文献[12]中设计了一种精巧的盲水印算法, 实现了无原图提取水印的方法。水印信号为二值 $\{-1, 1\}$ 随机序列, 嵌入的地方是小波变换后各层的 3 个高频区。用 $f_{k,l}(m,n)$ 表示第 L 层分解的高频分量, $k = h, v, d$ 分别表示水平、垂直和对角线分量。对每一层任意 m, n , 把 $f_{h,l}(m,n)$, $f_{v,l}(m,n)$, $f_{d,l}(m,n)$ 按从小到大的次序排列:

$$f_{k_1,l}(m,n) \leq f_{k_2,l}(m,n) \leq f_{k_3,l}(m,n) \quad (3-22)$$

$$\Delta = \frac{f_{k_3,l}(m,n) - f_{k_1,l}(m,n)}{2Q-1} \quad (3-23)$$

Q 是一个用户定义的自然数, 结果相当于把它们的间隔 $2Q-1$ 等分, 如图 3-7 所示。 Q 是一个人为给定的值。

中间的小波系数 $f_{k_2,l}(m,n)$ 落在由实线和虚线隔开的区间里。如果对应的二值水印为 -1, $f_{k_2,l}(m,n)$ 的值就被改为最近的实线对应的值; 对应的二值水印为 1, $f_{k_2,l}(m,n)$ 的值改为最近的虚线对应的值。

在提取水印时, 对受攻击后的图像小波变换后, 也做同样的排序, 如果实线离 $f_{k_2,l}(m,n)$ 近, 则此处的水印为 -1, 否则为 1。

水印嵌入的例子, $Q=4$

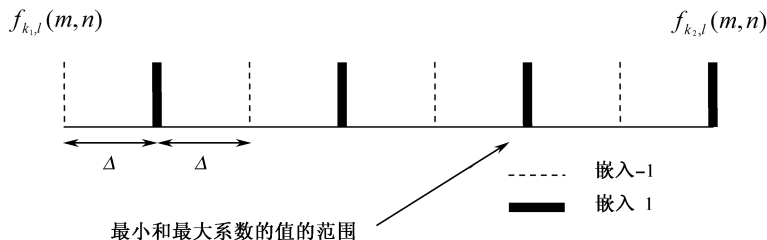


图 3-7 嵌入二值水印的量化过程

以上介绍的算法都基于多分辨率分析。多分辨率分析可以对信号进行有效的时频分解, 但是由于尺度的二进制变换, 高频段的频率分辨率较差, 在低频段时间分辨率较差。而小波包分析是将频带多层次划分, 即对多分辨率没有细分的高频部分进一步分解, 提高了时频分辨率, 为图像分析和处理提供了更丰富的选择。

文献[13, 14]都提出了基于小波包分析的数字水印算法。其中文献[13]的小波包分析里的“best basis”放弃了一般的极小化熵的选取方法, 而是按照能量标准, 以提高防压缩攻击

的能力。

几何攻击对很多扩频水印是致命的。Alghoniemy 等人提出了一种基于小波变换的抗几何攻击的算法^[15]。基本思想是找到受到放缩攻击的倍数和旋转的角度，然后反方向变化。用边缘标准偏离率 (ESDR) 表示放缩倍数，用平均边缘角度差异 (AEAD) 来估计旋转角度，ESDR 和 AEAD 都是在小波变换的基础上计算得来的。

3.5 Contourlet 域水印算法

二维可分小波是一维小波的简单张成，各向同性的性质导致方向选择差，不能有效利用数据本身特有的几何特征，并不是最优和最稀疏的函数表示方法。2002 年，M. N. Do 和 M. Vetterli 提出了 Contourlet 变换^[16]，它比小波变换具有更好的方向敏感性，具有多分辨率、多方向的特性，能够更稀疏地表示图像。

3.5.1 Contourlet 变换

图 3-8 (a) 给出了小波基逼近奇异曲线的过程，其基函数具有方形的支撑域，表现出各向同性的性质，仅能捕捉有限的方向信息（水平、垂直和对角方向），随着分辨率升高，尺度变细，最终表现为用“点”来逼近曲线。在逼近轮廓细节较多的图像时，误差衰减极为缓慢，最终表现为不能稀疏表示原图像。Contourlet 是一种基于图像的几何性变换，它将多尺度分析和方向分析分拆进行，有效地表示了轮廓和纹理丰富的图像。其支撑区间具有随尺度而长宽比变化的“长条形”结构，能有效地跟踪图像中的线奇异性 and 面奇异性特征。图 3-8 (b) 对同一奇异曲线给出了一种更为有效稀疏的表示，其基的支撑区域为不同规格的长方形，尤其在分辨率高的情况下，可使用少得多的基函数稀疏逼近曲线。与小波基方向支撑域的各向同性不同，条形的支撑区间与曲线方向的一致性是多方向的一种体现，称为各向异性。与小波变换相比，具有丰富基函数的 Contourlet 变换可以用更少的变换系数描述光滑边缘，并且将具有相同方向信息的奇异点汇集成奇异线或面。

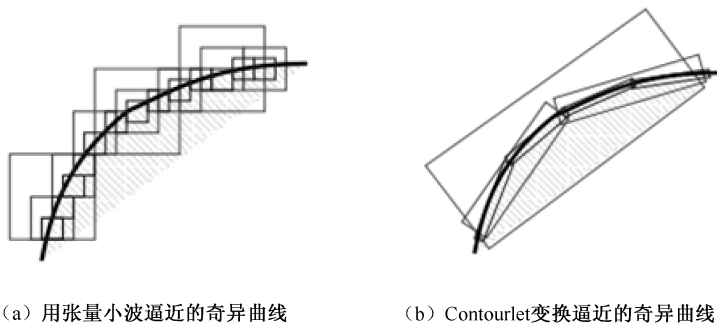


图 3-8 奇异曲线的不同逼近方式

Contourlet 变换也称塔形方向滤波器组 (Pyramid Directional Filter Bank, PDFB)，该变换将拉普拉斯金字塔 (Laplacian Pyramid, LP) 和方向滤波器组 (Directional Filter Bank, DFB) 进行组合，从而将多尺度分析和方向分析分开进行，对细小的有方向的轮廓和线段的表达有

着独有的优势。

图 3-9 显示了离散 Coutourlet 变换的滤波器组结构图，原始图像经 PDFB 结构多层分解得到多尺度方向的子带图像。

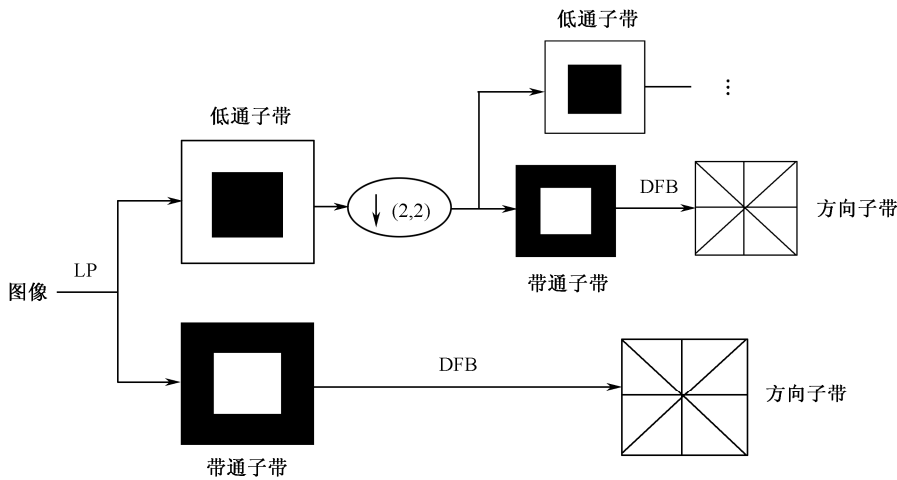


图 3-9 Coutourlet 方向滤波器组

Contourlet 变换的实现分为两个步骤：子带分解和方向变换。Coutourlet 变换选用 LP 滤波结构对图像进行尺度分解捕捉点奇异。接着由 DFB 滤波器将分布在同方向上的奇异点合成一个 Contourlet 系数。Coutourlet 变换的最终结果是用类似线段的基结构来逼近原始图像。Coutourlet 变换具有随尺度变换而长宽比变化的长条形结构，有很好的方向性和各向异性，在每个尺度所分解的方向灵活可变，该数目可以为 2^n ， n 为正整数。

图 3-10 (a) 和图 3-10 (b) 分别显示了 DWT 变换和 Contourlet 变换二阶分解的结果，其中 Contourlet 变换在最细分辨率上方向分解数为 8。

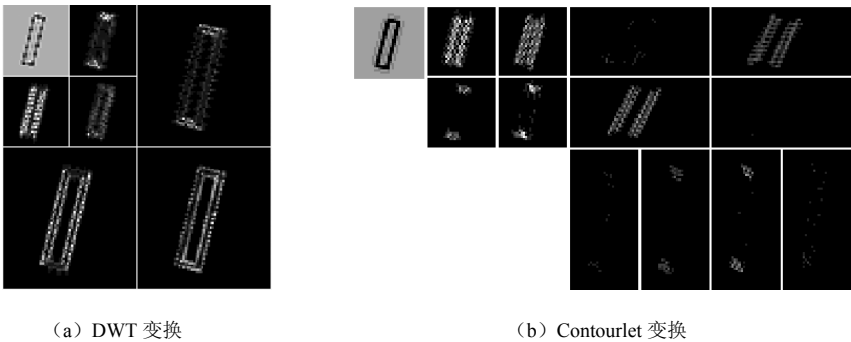


图 3-10 DWT变换与Contourlet变换二阶分解

与 DWT 变换相比，Contourlet 变换具有以下更好的特性^[17]。

1. 对图像更加灵活的多尺度描述

Contourlet 变换可以用方向滤波器组 DFB 将 LP 变换后的带通图像分解成指定个数的方向子带，能更好地提取图像纹理方向的分布。如图 3-10 (b) 所示，Contourlet 变换的方向子带更具体地体现了该方向上的轮廓和边缘分布，与小波变换的子带相比，其纹理方向性和分布

更加明确，在嵌入水印时可以更好地利用图像的纹理掩蔽特性。同时在 Contourlet 变换子带中，方向子带中的奇异点也代表了图像的重要特征系数，利用 Contourlet 变换，既可以提取出图像方向上的纹理特性，也可以提取出图像的重要系数。

2. 图像进行 Contourlet 分解后，系数之间是近似去相关的

如图 3-10 (b) 所示，能量主要集中在各尺度下方向子带的纹理和边缘位置上，同时系数变化是与大系数条件相关的，因此 Contourlet 子带系数的分布是具有非线性相关性的。图 3-11 给出了图像 Contourlet 变换后方向子带的系数直方图，体现了系数的概率分布特性：在零均值上方有尖锐的峰起，同时在峰起的两侧迅速衰减。显然，Contourlet 变换后的方向子带系数边缘概率分布可以用广义高斯模型拟合。假定嵌入的水印信息 $W(x,y)$ 由一个服从于均值为 0、方差为 1 的高斯分布的伪随机实数序列组成。把这样的一个水印嵌入 Contourlet 变换方向子带的过程，可以看成两个服从于同分布的信号叠加，这样既满足视觉上的不可见性，在数理统计上也是隐蔽的。

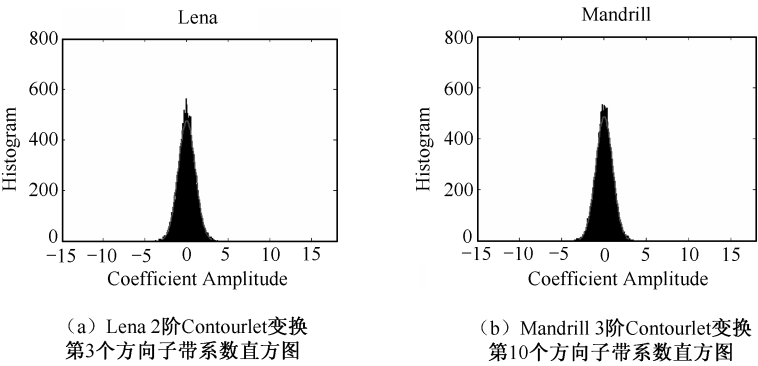


图 3-11 Contourlet变换方向子带系数直方图

3.5.2 基于 Contourlet 变换的水印算法

Contourlet 变换是一种多尺度几何变换，不仅具有小波变换的多分辨率和时频局部性，而且提供了高水平的方向性和各向异性，因此它在捕捉图像平滑的轮廓和几何结构上更有效。因此，基于 Contourlet 变换的水印算法有两个优势：第一，Contourlet 变换对细节的分析，使得水印的嵌入可以直接附着在图像的细节部分，也就是视觉不敏感区域，这使得水印具有更好的视觉掩蔽性；第二，图像细节部分在 Contourlet 域表现为数量级较大的系数，因此水印嵌入位置可以定为这些大系数，使得嵌入过程简洁有效。

其次，在非线性近似方面，使用相同个数的重要系数对图像进行重构时，小波变换用分离的点缓慢捕捉轮廓，Contourlet 变换可以迅速使用线条捕捉轮廓，使得水印在遭到攻击情况下，迅速恢复水印轮廓，提高水印提取的质量。

另外，随机噪声会产生类似真实边缘的小波重要系数，但不会产生 Contourlet 重要系数，该特点使 Contourlet 变换在图像去噪方面得到了广泛的应用。在水印算法中，Contourlet 变换域可以提高水印抗高斯噪声等攻击的能力，带来更强的鲁棒性。

目前已出现了一些基于 Contourlet 的水印算法,其中文献[18]将水印嵌入能量较大的 Contourlet 变换方向子带中,文献[19]把水印嵌入方向子带中绝对值大的系数上,而文献[20]是从方向子带中提取出具有显著特征点作为水印嵌入位置。这些水印算法都利用了 Contourlet 变换方向子带的纹理特性,能较好地协调鲁棒性与透明性。算法的不足是水印嵌入的自适应性较差,且抗几何攻击能力不足,与现有绝大多数图像水印方案一样,仅能够对抗常规的信号处理,而无法有效抵抗诸如旋转、缩放、平移、剪切等几何攻击。文献[17]提出了一种基于 Contourlet 变换域的以特征点作为模板的水印算法,水印被自适应地嵌入 Contourlet 变换域最高阶方向子带的相同带内坐标中纹理最丰富的位置,然后利用改进的 Harris-Laplace 算子从含水印的图像中提取出具有几何形变鲁棒性的特征点,将其作为模板。检测时首先利用特征点模板恢复几何形变图像,实现重同步后再检测水印。该方案具有很好的不可见性,且对常规信号处理和常见的几何攻击均具有很好的鲁棒性。

3.6 水印攻击

所谓水印攻击,就是对现有的数字水印系统进行攻击。通过检验其鲁棒性与安全性,分析其弱点所在及其易受攻击的原因而改进设计。这同传统密码学中的加密算法设计和密码分析是相似的。在对水印嵌入技术进行广泛研究的同时,部分学者致力于水印攻击技术的研究。与水印嵌入技术的发展类似,水印攻击技术也经历了一个快速发展的过程。可以说这两种技术是在互相斗争中同步发展起来的^[21]。

水印攻击方法可以分为四类:健壮性攻击、表达攻击、解释攻击和合法攻击。其中前三类可归为技术攻击,而合法攻击则完全不同,它是在水印方案所提供的技术特点或科学证据的范围之外进行的。在此,仅论述常见的前三类技术攻击方法和一些基本对策。

3.6.1 鲁棒性攻击

鲁棒性攻击以减少或消除数字水印的存在为目的,包括像素值失真攻击、敏感性分析攻击和梯度下降攻击等。这些方法并不能将水印完全除去,但可能充分损坏水印信息。为抵抗这类攻击,总体要求水印算法是公开的,算法的安全性应依赖于与图像内容有关或无关的密钥及算法本身的特性。

1. 像素值失真攻击

像素值失真攻击是指对图像像素值的修改,可分为信号处理攻击和分析攻击两种方法^[22]。

信号处理攻击是通过对水印图像进行某种操作,以削弱或删除嵌入的水印,而不是试图识别或分离水印。这种攻击包括线性或非线性滤波、图像压缩、添加噪声、图像量化、模数或数模转换等,造成像素值失真的四种基本攻击操作是外加噪声、幅值变化、线性滤波和量化,其他的攻击操作可看成这四种基本方式的有机组合^[23]。在这四种基本攻击操作中,线性相关检测对外加噪声以及归一化相关检测对幅值变化都是健壮的,而变阈值的优化检测方法,对线性滤波和量化处理比相关检测具有更好的健壮性。

对于不同的攻击操作,可采用鲁棒性较好的水印模型,如线性滤波对水印检测的影响依赖于加到每个载体频率上的水印信号能量的大小,因此可将水印模型设计成在滤波影响最小的频率上加入最大的水印能量。线性滤波实际上是信号与对称滤波器的卷积,并不影响 Fourier 系数的相位,所以将水印信号加到 Fourier 系数的相位上,不会受到这类线性滤波的影响。另外,使用扩频技术或在视觉显著的频率分量上嵌入水印,也能有效地抵抗多种像素值失真攻击。

对于使用优化的消除噪声的水印攻击方法,可采用满足功率谱条件的水印模式,使水印的功率谱与原图像的功率谱成比例;或将水印嵌入感知重要的频率分量上,如嵌入 DCT 变换的中低频系数上。攻击者为除去水印必须对水印图像施加很强的攻击,这时攻击过的图像一般不能使用。

分析攻击是通过分析水印图像来估计图像中的水印,然后将水印从图像中分离出来并使水印检测失败。常见的例子是合谋攻击,它有两种基本类型:其一是攻击者拥有同一个原图像嵌入了不同水印的拷贝,通过取所有拷贝的均值或仅从每个拷贝中取一小部分,可得到一个检测不到水印的原图像的近似值。其二是攻击者拥有嵌入了同一个水印的不同水印图像,对这些图像取均值并以这个均值作为嵌入水印的估计值,然后从水印图像中将这个估计值减去。它的一种变形是同一个水印重复嵌入一个数据的几个位置,再将这几个位置看成独立的,而合谋攻击依赖于获得很好的嵌入水印的估计值,借助于水印功率谱条件可削弱这类攻击。

2. 敏感性分析攻击

水印敏感性分析攻击^[24]的基本思想是:使用相关水印检测器寻找从水印检测区域到区域边缘的捷径,而该捷径可由检测区域表面的法线近似表示,并且该法线在检测区域的绝大部分是相对恒定的。敏感性分析攻击一般可分为三步实施。

① 对欲攻击的水印图像 I_w , 寻找一个非常接近相关检测区域边界的图像 I_{out} 。可通过多种方法改变图像 I_w 而获得图像 I_{out} , 如减少图像 I_w 的对比度或亮度的幅值, 使用无水印图像与水印图像 I_w 的线性组合, 使用水印图像 I_w 的平均值代替采样值。运用上述三种方法逐步改变水印图像 I_w 的失真程度, 直到不能检测到水印为止, 所得图像作为图像 I_{out} 。

② 找出图像 I_{out} 检测区域表面法线方向的近似值, 这是进行水印敏感性分析攻击的核心。文献[25]采用迭代技术估计检测区域表面法线, 每步迭代给图像 I_{out} 加上一个 N 维随机向量并记下相关检测结果, 如果检测到水印存在, 则将该随机向量加到法线的估计值上; 如果检测不到水印, 则从法线估计值中减去该向量。该法线方向估计值与图像 I_w 中水印的相关性是迭代次数的单调递增函数, 当相关性达到预定要求时停止迭代。

③ 对该法线进行缩放调整作为水印的近似值, 并将其从图像 I_w 中减去, 得到质量良好的检测不到水印的近似图像。

水印敏感性分析攻击的成功, 依赖于检测区域边界的法线可用于寻找越出检测区域的捷径。如果检测区域边界的曲率使在每一点的法线仅提供关于该捷径方向的极少信息, 则敏感性分析攻击在计算上是不可行的。因此构造具有这种性质的水印检测区域是一个需要关注的问题。

3. 梯度下降攻击

梯度下降攻击^[29]要求使用的水印检测器输出具体的检测值，而不仅是最终的二值判决结果（即是否检测到水印）。随着嵌入水印图像的缓慢改变，攻击者根据检测值的变化来估计水印图像检测统计量的梯度。这种攻击的基本思想在于：检测统计量下降最快的方向是越出检测区域的捷径。给定一个水印图像，可采用搜索策略确定检测统计量下降最快的局部梯度，图像沿该梯度方向可被某个量所改变。这种处理过程可以逐步迭代下去，直到在改变的图像中检测不到水印为止。

梯度下降攻击的成功依赖于如下假设：局部梯度指出了通向检测区域边界的捷径方向。这对许多检测统计量（包括线性相关和归一化相关）是必然的。为抵抗这种攻击，在检测区域内的检测统计量不应向边界方向单调下降，而应包含许多局部最小值，使得局部梯度方向不能提供确定越出检测区域边界的捷径的任何信息。

水印敏感性分析攻击和梯度下降攻击都是通过寻求从“水印存在”到“水印不存在”的边界所在，从而构造出不含水印的近似图像。它们虽然效果良好，但都需要具备水印检测器才能实施。

3.6.2 表达攻击

表达攻击是让图像水印变形而使水印存在性检测失败。与健壮性攻击相反，表达攻击实际上并不除去嵌入的水印，而试图使水印检测器与嵌入的信息不同步。当二者完全同步时，检测器能恢复嵌入的水印信息，但对同步处理的复杂性要求太高而不便于实用。为了战胜表达攻击，水印的检测算法应有与人交互的功能，或设计更复杂、更智能的包含所有表达攻击模式的检测器。

1. 置乱攻击

置乱攻击是指在将水印图像提交水印检测器之前，先对图像的像素值进行置乱，通过水印检测器之后再行逆置乱。这种置乱可以是像素值简单的行（或列）的置换，也可以是比较复杂的随机置乱，置乱程度与使用的检测策略有关。最著名的置乱攻击是马赛克攻击^[26]，该攻击方法的目的是挫败 Webcrawler。它将嵌入了水印的图像分割成许多检测不到水印的小方块，这些小方块在 Web 页上按相应的 HTML 标记重新组装起来。Webcrawler 只能查看每个图像小块，但由于这些小块太小而无法容纳水印数据，所以 Webcrawler 无法发现水印。对付这类攻击的一种策略是检测算法与人相结合。

2. 同步攻击

许多水印技术对同步性非常敏感，要求在检测水印之前，嵌入了水印的图像必须正确对齐。攻击者可在保真度的约束下，通过对图像的几何变形来干扰这种同步性，使得水印虽然存在但却检测不出来。

引起失同步的这些几何变形可以是简单的平移、旋转、缩放，或较复杂的图像剪切、水平翻转、行（或列）删除，以及随机几何变形（如直方图拉伸、均衡、非线性扭曲等），甚至

是某些几何变形的组合。攻击者可利用水印攻击软件 Unzign、Stirmark 等实施攻击。Unzign 引入了局部像素抖动,在对空域水印方案攻击时很有效;Stirmark 引入了全局和局部两种几何失真;Checkmark 可以进行扭曲、模板移除等攻击。此外,还可利用水印同步方案的知识设计专用的攻击方法。

水印系统设计者所能采取的对策通常是预见可能的攻击方式,提高水印系统对同步攻击的健壮性。常用的方法包括同步模板登记技术、自相关函数方法、不变水印技术、内在同步技术等。自相关函数方法和同步模板登记技术要求确认几何变形和逆转变形后的水印检测必须是成功的,缺一不可;内在同步技术要求显著特征点在检测时能可靠地提取出来,但有些几何变形会影响显著特征点与图像的相对位置,从而使水印无法检测,这在设计水印系统时必须加以注意。

对于一个成功的表达攻击而言,并不需要削弱或除去水印,因此它几乎不影响图像质量,这是健壮性攻击和解释攻击所无法比拟的。正由于表达攻击没有削弱或除去水印,当使用更复杂、更智能化的水印检测器时,很可能检测到图像中的水印,这是表达攻击的一个致命的弱点。

3.6.3 解释攻击

在一些水印方案中可能存在对检测出的水印具有多种解释。解释攻击包括拷贝攻击、可逆攻击等,它使数字水印的版权保护受到了挑战。潜在的解决方法是构建与图像内容相关的数字水印。

1. 拷贝攻击

拷贝攻击^[27]是从嵌入水印的图像中估计出水印并拷贝到目标图像的其他图像中。拷贝的水印要自适应于目标图像,以保证其不可察觉性。使用拷贝攻击在目标图像中生成一个有效的水印,这既不需要算法知识又不需要水印密钥知识。拷贝攻击分为三步进行:

- ① 找出图像中水印的估计值;
- ② 处理该估计值,使得水印能量最大化并满足不可感知性要求;
- ③ 将处理后的水印估计值嵌入目标图像,得到伪造的水印图像。

2. 可逆攻击

可逆攻击^[28]基于大多数水印方案的嵌入算法是可逆的和多数水印嵌入是健壮的这一事实,攻击者将水印嵌入过程逆过来使用。可逆攻击对盲水印系统同样适用,可建立一个类似噪声但与发布的图像具有很高相关性的伪造水印并实施攻击,这样的水印可通过提取和改变发布图像的某些特征来构造。攻击者从发布的图像中减去伪造的水印便可建立一个伪造的原图像,从而使水印检测陷入死锁,造成图像所有权的模糊性。

水印嵌入算法必须设计成不可逆的。方法是使水印依赖于图像的内容,如使用原图像的单向杂凑值作为伪噪声发生器的种子生成水印,这时攻击者要伪造一个原图像在计算复杂性上是不可能的;也可利用数字签名技术,将嵌入的水印及签名连同原图像的签名一起嵌入,这种方法可证明是安全的。

拷贝攻击和可逆攻击都属于在协议层上的攻击，会对水印的许多应用造成严重损害。在版权保护方面，可逆攻击的威胁要大得多，因为任何人都可声称他对访问过的任何水印图像拥有所有权；拷贝攻击在这方面的应用是作者盗用某名人的名义，出售自己的作品以牟利。在有关身份认证的应用中，拷贝攻击所造成的威胁是重大的，使得用户无法根据水印的检测结果确定作品来源的真实性。

从以上分析可以看出，健壮性攻击将对水印造成实质性的损害，遭受这类攻击的水印是难以检测或恢复的；表达攻击是水印方案面临的公开问题，目前仍然缺乏有效的对抗策略，只能通过预见可能遇到的具体攻击方法进行预防，由于它不影响水印的存在性，使用更先进的检测器可能检测到攻击过的水印，解释攻击破坏了水印应用的基础，攻击的是水印必须具有的唯一性解释，但在采取相应的措施后这种攻击是难以实施的。

3.7 Stirmark 基准测试程序

3.7.1 Stirmark 概述

基准测试程序是用于测试硬件或软件性能的程序。数字水印基准测试程序作为一种软件基准测试程序，是通过对水印作品应用各种变换以评价标记的鲁棒性和安全性的特定计算机程序。当前的数字水印基准测试程序包括 Stirmark 基准测试程序、Optimark 基准测试程序、Checkmark 基准测试程序和 Unzign 基准测试程序等，每种水印基准测试程序都各有特点，Stirmark 是最为流行的。

Stirmark 基准测试程序^[29]是一种通用的水印算法鲁棒性测试程序，由英国剑桥大学的 Petitcolas 等人开发。它是采用软件方法来实现对水印载体图像进行各种攻击，从而实现对图像水印算法的鲁棒性测试，从 1997 年 11 月开始免费提供。

对于给定的水印算法，Stirmark 可以从多方面测试水印算法的鲁棒性，用于测试的攻击手段包括线性滤波、非线性滤波、剪切/拼接攻击、同步破坏攻击等，同时还为用户保留了自定义测试方法的接口，用户可以很方便地在 Stirmark 中定义自己的测试例程。

StirMark 基准测试程序具有以下特征^[30]：

- ① 使用用户提供的动态链接库作为水印标记方案函数；
- ② 对在 INI 文件中指定的文件夹所包含的所有媒体文件执行测试；
- ③ 每个测试都可自定义，且测试参数可在 INI 文件中设置；
- ④ 在 LOG 文件中导出量化测试结果，同时在输出文件夹中导出失真图像；
- ⑤ 用户可容易地编写自定义的测试和攻击。

人们可以以水印检测器能否从遭受攻击的水印载体中提取或检测出水印信息来评定水印算法的抗攻击能力。如 Stirmark 可对水印载体进行重采样攻击，它首先模拟图像用高质量打印机输出，然后再利用高质量扫描仪扫描，得到其图像在这一过程中引入的误差。

另外，Stirmark 还可以对水印载体图像进行几何失真攻击，即它可以以几乎注意不到的轻微程度对图像进行拉伸、剪切、旋转等几何操作。Stirmark 还通过应用一个传递函数，来模拟非线性的 A/D 转换器的缺陷所带来的误差，这常见于扫描仪或显示设备^[31]。

3.7.2 用户 API 接口

Stirmark 基准测试程序使用 C++语言编写，为客户程序提供了一个标准的接口和一套基于目录结构和配置文件的测试配置方法。在使用 Stirmark 基准测试程序测试数字水印算法之前，必须先根据 Stirmark 基准测试程序用户 API 接口标准定义水印算法，同时将其打包为动态链接库，之后根据 Stirmark 基准测试程序配置方法为水印算法的测试定义，来测试图像库和测试列表。然后就可以使用 Stirmark 基准测试程序测试水印算法了。

Stirmark 基准测试程序的 API 接口定义了水印方案参数结构 SMBSchemePars、图像信息结构 SMBImage 和 3 个主要的 API 接口函数：GetSchemeInfo 函数、Embed 函数和 Extract 函数（表 3-2）。通过这 3 个 API 函数，Stirmark 基准测试程序就可以定义一致的水印算法接口标准^[35]了。

表 3-2 水印方案的 API 接口函数

获取水印方案的信息： _declspec (dllexport) ErrorNum GetSchemeInfo_Lib (int in_nInfo, unsigned char * out_pbData, unsigned int *inout_pcbData);
水印方案的嵌入算法： _declspec (dllexport) ErrorNum Embed_Image_Lib(const SMBImage in_imgOriginal, /* 原始图像 */ SMBImage *out_pimgTarget, /* 要输出的水印图像*/ const SMBSchemePars * in_pPars /*嵌入参数 */);
水印方案的提取算法： _declspec (dllexport) ErrorNum Extract_Image_Lib(const SMBImage in_imgTest, /*待测试的图像*/ const SMBImage in_imgOriginal, /*原始图像或空图像*/ SMBSchemePars* inout_pPars/*提取和输出参数*/);

3.7.3 配置测试方案

Stirmark 基准测试程序通过目录结构和配置文件实现对包括图像集和测试集的测试方案的配置。

Stirmark 基准测试程序目录结构如表 3-3 所示。Stirmark 根目录由执行文件（Bin）目录、配置文件（Profiles）目录和媒体集（Media）目录组成。执行文件目录又由基准测试程序（Benchmark）目录和水印算法库文件（Libraries）目录构成，其中 Benchmark 目录中包含了 Stirmark 基准测试程序本身和测试输出的日志文件。配置文件目录包含了 Stirmark 基准测试程序的配置文件，配置文件在此文件夹测试时创建，并通过 Stirmark 基准测试程序

的命令行指定。媒体集文件夹包含了输入（Input）文件夹和输出（Output）文件夹，分别对应于 Stirmark 基准测试程序的输入媒体集和经过测试（变换）的输出媒体集。在输入和输出文件夹下又包含图像（Images）文件夹和声音（Sound）文件夹等，分别对应于不同的媒体类别（当前 Stirmark 基准测试程序只支持图像水印算法的测试，对于其他媒体仅仅保留了扩展接口）。

表 3-3 目录结构和媒体集

/ Bin /			Benchmark/	基准程序本身
			Libraries/	用户的库
Profiles /				初始化配置文件
Media/	Input/	Images/	MyFolder1/ MyFolder2/ ...	按照某标准存储的样本
		Sounds/	MyFolder1/ MyFolder2/ ...	
	Output/			相同于 Input 的结构

1. 测试图像集

为了给 Stirmark 基准测试程序指定使用的图像集，需要将测试图像添加到相应的文件夹中，然后在配置文件中指定此测试图像集。

为了在配置文件中指定测试图像集，需要在[ImageFolders]节点下将 Foldern 指定为测试图像集文件夹相对于输入文件夹的相对路径，n 是文件夹序号。例如：

```
[ImageFolders]
Folder1=Images/MyFolder1
Folder2=Images/MyFolder2
```

有了这些信息，Stirmark 基准测试程序就知道了它需要对包含在 Media/Input/Images/MyFolder1 和 Media/Input/ Images/MyFolder2 中的所有图像执行测试，同时将变换后的图像分别存储在 Media/Onput/ Images/MyFolder1 和 Media/Onput/Images/MyFolder2 文件夹中。

2. 测试列表

Stirmark 基准测试程序包含了大部分常用到的测试，如 JPEG 压缩、几何变换等，同时也保留了相应的扩展接口，即可以让用户自己来定义新的测试。Stirmark 基准测试程序通过配置文件来定义测试和设置测试参数。

为了给 Stirmark 基准测试程序指定要使用的测试集，需要在[Tests]节点下将 Testsn 指定为相应的测试名，在这里 n 是测试序号。同时使用测试名新建一个节点，以在此处指定此测试的参数设置。例如：

```
[Tests]
Test1= Test_PSNR
Test2= Test_Rotation
```

```
[Test_PSNR]
; 水印嵌入强度参数 10, 20, ..., 80

start=10
end=80
step=10
```

3.7.4 执行测试程序

Stirmark 基准测试程序使用三个参数作为命令行选项：数字水印算法库文件、评估配置文件（INI 文件）和日志文件（LOG 文件）。它假定特定的文件夹结构已经生成。各参数的含义如下。

- ① 数字水印算法库文件：定义水印算法的 Embed/Extract/GetSchemeInfo 函数。
- ② 评估配置文件：包含了运行测试的参数，同时也可以为不同的数字水印应用程序使用不同的评估配置。
- ③ 日志文件：用于导出结果，其中包含了水印算法测试的诸如鲁棒性量度和不可感知性量度的量度值。

这里给出 Stirmark 基准测试程序的一个使用示例：

```
StirMarkBench myDLL.dll myINIFile.ini myLOGfile.log
```

Stirmark 基准测试程序默认的设置是 EmbedDLL.dll，SMBsettings.ini 和 SMBReport.log。

参考文献

[1] R. G. van Schyndel, A. Z. Tirkel, C. F. Osborne. Digital Watermark. International Conference on Image Processing,1994 (2): 86-90.

[2] I. J. Cox, J. Kilian,T. Leighton, et a1. Secure Spread Spectrum Watermarking for Images, Audio and Video, in Proceedings of IEEE International Conference on Image Processing, 1996 (3): 243-246.

[3] M. Kutter, S. K. Bhattacharjee, and T. Ebrahimi. Towards Second Generation Watermarking Schemes, in Proceedings of International Conference on Image Processing, 1999 (1): 320-323.

[4] 李雷达. 水印抗几何攻击理论及应用研究. 西安电子科技大学博士学位论文, 2009.

[5] W. bender, D. Gruhl, N.Morimoto, A. Lu. Technique for Data Hiding. IBM System Journal, 1996, 35(3-4): 313-336.

[6] B. Tao, B. Dickinson. Adaptive watermarking in the DCT domain, International Conference on Acoustics, Speech, and Signal Processing(ICASSP), Munich, Germany, 1997(4). 2985-2988.

[7] C. I. Podilchuk, W. J. Zeng. Image-adaptive watermarking using visual model, IEEE Journal on Selected Areas in Communications, 1998, 16(4): 525-539.

- [8] M. Barni, F. Bartolini, V. Cappellini, A. Piva. A DCT-domain system for robust image watermarking. *Signal Processing*, 1998, 66(3): 357-372.
- [9] 张冠男. 小波域自适应盲水印算法和半色调图像水印算法的研究. 吉林大学硕士学位论文, 2005.
- [10] M. Corvi, G. Nicchioyi. Wavelet-based image watermarking for copyright protection, In *Scandinavian Conference on Image Analysis SCIA'97*, 1997.6.
- [11] D. Kundur, and D. Hatzinakos. A Robust Digital Image Watermarking Method Using Wavelet-Based Fusion, in *Proceedings of the International Conference on Image Processing*, vol.1, Santa Barbara, California, 1997.10.544-547.
- [12] D. Kundur, and D. Hatzinakos. Digital Watermarking Using Multiresolution Wavelet Decomposition. In *International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, Seattle, WA, USA, 1998.5, 2969-2972.
- [13] J. L. Vehe, A. Manoury. Wavelet packet based digital watermarking. In *Proceedings of the 15th International Conference on Pattern Recognition*, 2000.
- [14] A. Pommer, A. Uhl. Wavelet packet methods for multimedia compression and encryption, In *Proceedings of the 2001 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Victoria, Canada, 2001.
- [15] M. Alghoniemy, A. H. Tewfik. Geometric Distortions Correction in Image Watermarking, In *Proceedings of SPIE on Security and Watermarking of Multimedia Contents II*
- [16] M. N. Do, M. Vetterli. Contourlets: a new directional multiresolution image representation, *Signals, Systems and Computers*, 2002, (1): 497-501.
- [17] 楼偶俊, 王钺旋. 基于特征点模板的 Contourlet 域抗几何攻击水印算法研究, *计算机学报*, 2009, 32(2): 308-317
- [18] 李海峰, 宋巍巍, 王树勋. 基于 Contourlet 变换的稳健性图像水印算法. *通信学报*, 2006, 27(4): 87-94.
- [19] M. Jayalakshmi, S. N. Merchant, B. D. Uday. Digital watermarking in contourlet domain, *18th International Conference on Pattern Recognition. Hong Kong, China*, 2006, 3: 861-864.
- [20] Bouzidi Ali, Baaziz Nadia. Contourlet domain feature extraction for image content authentication, *Proceedings of the 2006 International Conference on Intelligent Information Hiding and Multimedia Signal Processing. Victoria, Canada*, 2006: 202-206.
- [21] 刘春庆, 王执铨, 戴跃伟. 常用数字图像水印攻击方法及基本对策, *控制与决策*, 2004, 19(6): 601-606.
- [22] 杨义先, 钮心忻, 任金强. 信息安全新技术. 北京: 北京邮电大学出版社, 2002.
- [23] I. J. Cox, M. L. Mill, J. A. Bloom. *Digital Watermarking*, San Francisco: Morgan Kaufmann Publishers, 2001. 241-316.
- [24] T. Kalker, J. P. Linnartz, M. V. Dijk. Watermark estimation through detector analysis. *IEEE International Conference on Image Processing*. Los Alamitos, 1998, 425-429.
- [25] V. Solachidis, A. Tefas, S. Tsekeridou, et al. A benchmarking protocol for watermarking methods. *Proceedings of IEEE International Conference on Image Processing*, 2001: 1023-1026.

- [26] F. A. P. Petitcolas, R. Anderson, M. G. Kuhn. Information hiding: A survey. Proceedings of the IEEE, 1999, 87 (7): 1062-1078.
- [27] M. Kutter, S. Voloshynovskiy, A. Herrigel. The watermark copy attack. Proceedings of the SPIE, San Jose, 2000: 371-380.
- [28] S. Craver, N. Memon, B. L. Yeo, et al. Resolving rightful ownerships with invisible watermarking techniques : Limitations , attacks and implications[J]. IEEE J on Selected Areas in Communications ,1998 ,16 (4): 573-586.
- [29] F. A. P. Petitcolas, R. J. Anderson, M. G. Kuhn. Attacks on Copyright Marking Systems. 1998.
- [30] 张卫, 高政. StirMark 基准测试程序在数字水印方案评价中的应用. 电视技术, 2004, 8: 75-77
- [31] 易开祥, 石教英, 孙鑫. 数字水印技术研究进展. 中国图象图形学报, 2001, 6(2): 111-117

数字指纹技术

数字水印向数字产品中嵌入版权拥有者的一些信息，当发生争议时能够有效确认版权归属，作为解决版权的证据。但是，授权用户可能基于某种目的将已获得的受保护内容或密钥以某种方式分发给非授权用户，这些非法分发的授权用户也称叛逆者。数字水印是无法追查到叛逆者的，为了解决这个问题，数字指纹技术应运而生了。

数字水印嵌入的是版权信息，对相同的作品嵌入的水印信息是相同的。而数字指纹是在数字产品中嵌入与用户有关的信息，嵌入的内容对不同购买者是不同的。一旦发现非法复制，内容提供者（也称发行商）就会从数字产品中提取出用户的数字指纹，并根据这些信息对叛逆者进行跟踪。

指纹技术最早的文献是 Wagner 在 1983 年发表的题为“Fingerprinting”的文章^[1]，文章介绍了指纹的思想和一些术语，对指纹技术进行了分类，并给出了一些指纹技术应用的例子。Wagner 首先扩展了指纹的概念，认为指纹应该是普遍存在的，任何可能被滥用的对象都应该为其添加一个指纹，使得在它被滥用之后能够根据指纹识别该对象的所有者。

文献[2]中，Boneh 和 Shaw 首先提出了嵌入假设，并提出了一种抵抗多用户合谋攻击的指纹编码方案，该方案成为了离散指纹编码方案中的经典，在随后的多篇文献中得到了应用、改进和补充。这类编码方案的缺点在于编码长度较长并且译码算法的效率较低。

另外一个经典指纹方案是文献[3]中提出的连续指纹方案。该文中 Cox 等用独立随机的正态分布序列作为指纹码字，采用相关值检测跟踪合谋用户。文献[4]对这类指纹编码方案的译码算法做了改进，提高了跟踪效率。

叛逆者跟踪是与数字指纹相关的领域。1994 年，B. Chor 等人首先在文献[5]中介绍了数字指纹在卫星电视中的应用，提出了广播加密体制中的叛逆者跟踪。文中提出了三种方案来对抗多个不诚实用户的合谋攻击，其思想成为了后来许多抗合谋数字指纹方案的基础。

数字指纹技术具有广泛的应用环境和广阔的应用前景，它可用于在线出版业方面，如电子图书馆的构建；随着数字电视和数字广播的发展，可应用于 DVB（Digital Video Broadcast）、VOD（Video On Demand）等环境下的付费数据的保护。数字指纹技术与数字水印、数据加密、数字签名等技术一样是数字内容保护的重要手段。

4.1 数字指纹的基本概念

4.1.1 数字指纹的系统模型

数字指纹体制主要由两部分构成（图 4-1）^[6]，一部分是用于向数据复制品中嵌入指纹并对带指纹复制品进行分发的分发体制，另一部分是用于实现对非法分发者进行跟踪并审判的跟踪体制。以上两部分通过发行商、用户（也可能是登记中心、审判者等实体）之间的一系列协议实现。因此数字指纹体制也可以分为算法和协议两部分。其中，算法包括指纹的编码和解码、指纹的嵌入和提取以及复制品的分发策略等内容，而协议部分则规定了各实体之间如何进行交互以实现具有各种特点的复制分发和跟踪体制（如实现用户的匿名性等）。

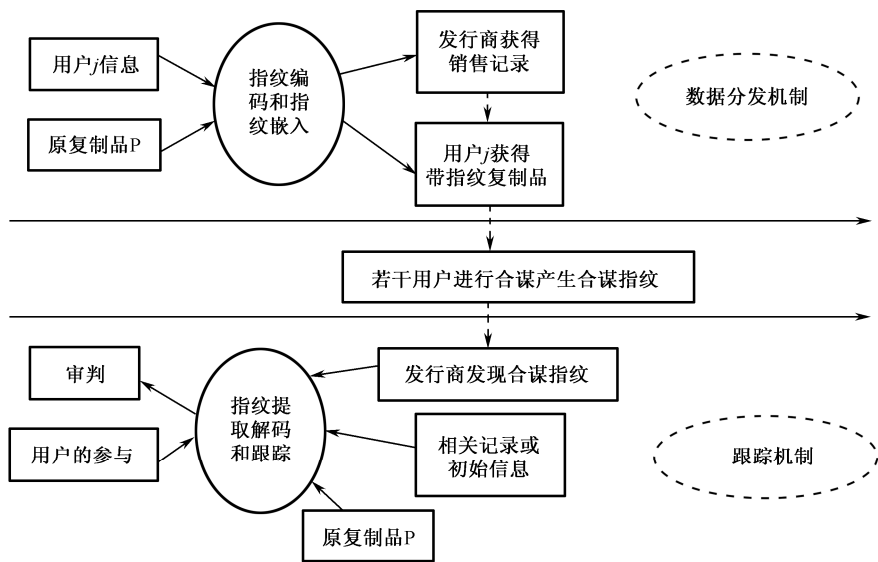


图 4-1 数字指纹系统模型

数据分发体制处理数据分发过程，包括数字指纹编码和指纹嵌入。用户 j 的信息由用户提供或由其与发行商（或登记中心等实体）通过一系列交互后生成。它通常包括用户的身份信息及该次购买过程的描述信息。有关用户 j 的信息将被按照一定规则进行编码并嵌入发行商要出售的原复制品中。用户直接得到带有其指纹的复制品或由发行商将带有指纹的复制品发放给用户，同时发行商和用户得到有关交易记录。

跟踪体制处理盗版跟踪过程，包括对盗版数据进行指纹提取和指纹跟踪。不诚实的用户可能会直接分发他所得到的复制品，也可能与其他用户联合获得新的复制品后再分发。无论是哪种情况，非法分发的复制品中都会留下参与非法活动用户的指纹信息。一旦发行商发现了非法复制品，他将运用相应的指纹提取及指纹解码技术，并运用跟踪算法跟踪非法分发者。

一般来说,只要发行商能够成功地跟踪出一个非法分发者,就认为该跟踪算法是成功的;如果该跟踪过程没有识别出任何一个盗版者或将一个无罪用户指证为盗版者,则认为该跟踪算法是失败的。指纹的编码算法以及合谋人数大小是影响跟踪成败的关键因素。

4.1.2 数字指纹方案的基本要求

数字指纹方案通常应满足以下几项基本要求^[7]。

1. 保真性

嵌入指纹后的数据复制品相对于原复制品,其质量不应降低,这实际上是信息隐藏方案的基本要求。

2. 鲁棒性

嵌入的指纹信息要能够抵抗可能受到的处理、操作甚至恶意攻击,使得提取出的信息足以跟踪出非法分发者。鲁棒性要求攻击者不能对指纹进行随意修改,其理想目标是使攻击者无法在不破坏原复制品的情况下伪造出一个新的可用复制品。

3. 嵌入量

因为嵌入的内容要实现用户攻击后能留下足够的信息使发行商进行跟踪,因此要求有足够的信息量。

4. 合谋容忍性

这是对数字指纹的一个关键要求。通常从以下两个方面考虑合谋容忍性:

① 在一定的合谋人数下,发行商能够确定出至少一个非法分发者,该人数称为合谋安全尺寸;

② 无论合谋人数的多少(即使超过了上述尺寸),无辜购买者也不能受到指控。

5. 效率

要求带指纹复制品的生成算法和跟踪算法的实现具有很好的效率。

以上是对数字指纹体制的若干基本要求,此外还有其他的一些要求,如实现用户的不可否认性和用户的匿名性等。针对不同的需求环境,对指纹体制的各项要求侧重点也会有所不同。在数字指纹体制中,具有较强鲁棒性的指纹嵌入算法,具有抗合谋攻击能力的编码和跟踪方案,以及有效、快速的协议实现是决定指纹方案的安全性和效率的关键环节。一般,嵌入方法可以借鉴数字水印中的嵌入技术,近年来已出现了各种各样的具有一定鲁棒性的嵌入算法,对指纹编码(及相应的跟踪算法)和指纹协议的研究是近年来国际上对数字指纹的研究热点。

4.2 数字指纹编码

数字指纹研究的热点之一集中于数字指纹的编码，通过抗合谋编码来抵抗合谋攻击。目前数字指纹编码和跟踪大多基于组合论的方法。

4.2.1 合谋攻击

合谋攻击，是几个盗版者联合起来，通过对各自的复制品进行比较，定位出部分标记的位置。然后通过综合所有原始数据复制品制造出一个新的数据复制品，试图隐藏自己的身份。

当盗版用户合谋产生出非法复制品，他们将采用不同的策略。这种策略一般是使他们自己被抓获的可能最小，或使陷害其他无辜用户的概率最大化。数字指纹的编码、跟踪方案都应充分考虑合谋者所采用的不同策略。一般的抗合谋方案也都是针对一定的合谋策略的，它在一定范围内有效，当合谋策略超出这个范围，整个指纹方案就可能被攻破。因此，在设计整个数字指纹方案时要充分考虑到合谋攻击的不同策略和方法。

合谋用户在可发觉位置上，依照某种策略选择他们码字中的某一个，从而生成新的非法指纹。这种策略可能是随机地选择 $\{0, 1\}$ 中的码字，也可以是简单意义上的代数运算、逻辑数学中的逻辑运算、统计数学中的统计运算，还可以是综合这些运算的组合运算等。下面简单介绍几种常见的合谋攻击。

1. 逻辑与

即逻辑运算中的“与”运算，合谋指纹由所有参加运算的指纹码字按位“与”得到。在“逻辑与”情况下，码“0”与任意码元相“与”，结果码字都为“0”；码“1”只有与码“1”相“与”时才为“1”。

2. 逻辑或

即逻辑运算中的“或”运算，合谋指纹由所有参加运算的指纹码字按位“或”得到。在“逻辑或”情况下，码“1”与任意码元相“或”，结果码字都为“1”；码“0”只有与码“0”相“或”时才为“0”。

3. 简单平均

即为算术运算或统计运算中的求取平均值。这是一种比较简单，合谋用户也比较容易实现的合谋策略。其中码字的平均值代表了该位置上选择“1”的概率，某个位置上码字“1”越多，其均值越接近于“1”，反之则越靠近“0”。

4. 随机选取

合谋用户比较他们的复制品，在不相同的位置上，随机地选择两种方式中的一种。合谋策略可以看成在可发觉位置上，选取“1”的概率为 p ，选取“0”的概率为 $1-p$ 。

5. 最大、最小策略

最大策略中,合谋用户在可发觉位置上选择在该合谋集中该位置上出现次数较多的码元。最小策略中则相反。当某位置上各码元出现次数相同时,则任选其一。当合谋用户数为2时,相当于随机选取策略。

由于数字指纹方案要对抗用户的合谋攻击,通常发行商会对用户的指纹进行编码,以增加该指纹方案的合谋容忍能力,这种编码称为合谋容忍编码。从码字的分布而言,指纹编码方案可以分为连续指纹方案和离散指纹方案。

抗合谋攻击能力是数字指纹系统的一个很重要的性能指标,因此指纹编码要充分考虑生成指纹码字的抗合谋能力。若一个数字指纹体制能够抵抗合谋攻击,则称该指纹编码方案是合谋安全的(Collusion-secure)。下面介绍几种抗合谋攻击的指纹编码方案。

4.2.2 连续指纹编码

连续指纹编码方案中,用户码字中的每一个码元取自一个连续的集合,如一个实数区间。文献[3]中的方案是此类编码方案的典型代表。该文中Cox等用独立随机的正态采样序列作为要嵌入的水印信息,当用做指纹时,为每个用户选取不同的采样序列,序列间是独立的。这里指纹的取值不限于离散的整数值,而是服从正态分布 $N(0, 1)$ 的随机实数序列 X 。跟踪时发行商从非法复制品中提取出嵌入信息 X' ,将其与 X 做相关检测,如果相关值大于某一个门限值,则认为非法复制品中含有该指纹 X 。称这种体制是CKLS体制。实验显示:当合谋者采用平均攻击生成盗版复制品时,从盗版复制品中提取出的信息与合谋者指纹的相关性明显高于与无关指纹的相关性,即CKLS方案具有较好的合谋容忍性。文献[6]为CKLS体制构建了明确的数学模型:设指纹信号是 n 维向量 X (其分量独立取自于正态分布 $N(0, a^2)$),设原复制品是 n 维向量 V ,指纹嵌入过程是 $V' = V + X$,水印检测方法就是CKLS体制中的相关检测法。连续指纹编码方案主要基于以下三个假设。

假设 4-1: 代表原始复制品的向量 V ,其分布独立取自于正态分布(不失一般性假设为 $N(0,1)$)。

假设 4-2: 假设攻击者不知道水印的检测器,即他不能用水印检测器检验自己的攻击效果。

假设 4-3: 原复制品 V 与嵌入信息后的复制品 V' 的相似性(在视觉上的不可区分性)可以用下式来衡量: $\|V - V'\| \leq \delta \sqrt{n}$ (其中 n 是指纹序列的长度, $\|\cdot\|$ 表示欧几里德范数, $\|X\| = \sqrt{X \cdot X}$, $X \cdot X$ 表示内积)。发行商和攻击者产生的可用复制品必须满足相似性准则。

在文献[8]中,作者指出上述方案是 c -安全的,其中 $c = O(\sqrt{n/\ln N})$, N 是不同指纹的个数。即 c 个合谋者生成的复制品 V^* ,或者与 V 不满足相似性准则,或者发行商可以从合谋复制品中检测出至少一个合谋者,且无辜用户以较高的概率不受合谋者的陷害。实际上, V 的各个分量的独立同分布特性不一定能够得到保证,因此文献[9]对假设4-1进行了放宽,并得出 $c = O(\sqrt{n/\ln N})$ 个合谋用户可以使任何CKLS型的方案不再是安全的,其中 n 是复制品的长度。同时也指出了当用户数目 $N = n^{O(1)}$ 时,文献[8]中的合谋安全尺寸 $c = O(\sqrt{n/\ln N})$ 是最优的。值得注意的是,上述方案中在对用户进行跟踪时的计算复杂度是 $c = O(N)$ (其中 N 是

用户的总数目)。当用户数目较大时,跟踪效率较低。因此文献[10]对 CKLS 型指纹编码方案的跟踪算法效率进行了改进。作者利用 Reed-Solomon 纠错编码,并采用级联码的思想,以 CKLS 编码作为内码,构造了改进的 CKLS 方案。该方案以适当减小合谋尺寸为代价,使指纹跟踪算法的效率从与用户数目 N 呈线性关系提高到与 $\log N$ 呈多项式关系。

尽管以上利用服从正态分布的随机序列作为指纹的方法具有较好的合谋容忍性,但如何将 CKLS 模型中的三个假设进行放宽还有待进一步研究。

4.2.3 C-安全码

1995 年 D. Boneh 和 J. Shaw 发表了“Collusion-Secure Fingerprinting for Digital Data”一文^[2],这是关于数字指纹编码方案的一篇经典文献。该文中的编码方法与使用的数据嵌入算法无关,只要其能够满足嵌入假设(Marking Assumption)。基于该假设,运用级联码的思想并在编码方案中引入随机性。作者给出了带有错误概率 ε 的 n -安全的 (l,n) 码,其指纹码字长度 l 与用户数目 n 的对数及合谋容忍尺寸 c 的四次方成正比,即 $l = O(c^4 \log(n/\varepsilon) \log(1/\varepsilon))$,发行商能以大于 $1-\varepsilon$ 的概率进行跟踪。该文的编码及跟踪思想在后来多篇关于指纹体制构造的文献中得到了应用。

假设 4-4: 嵌入假设。

合谋用户通过对比他们的复制品,只能在复制品相异之处发现指纹并进行修改。对于没有发现不同的指纹所在之处,除非将复制品变得无用,他们不能对该处的指纹进行修改。对于通过对比复制品发现的指纹比特,假设合谋者只可能将其修改为 $(0, 1, ?)$ 3 种状态(? 表示无法识别是 0 还是 1)。

作者首先给出了带有错误概率 ε 的 n -安全的 (l,n) 码,码本记为 $\Gamma_0(n,d)$ 。令 c_m 是高度为 n 的一个列,其中前 m 个元素为 1,其余为 0,则 $\Gamma_0(n,d)$ 包括所有列 c_1, \dots, c_{n-1} 。每个重复 d 次, d 将决定错误概率 ε 。对上述列进行排列后的每一行称为一个码字,共 n 个码字,码长为 $l = d(n-1)$ 。 c_i 所占的 d 个比特位记为 B_i 。例如,针对 5 个用户的码本 $\Gamma_0(5,3)$ 为:

a_1	111	111	111	111
a_2	000	111	111	111
a_3	000	000	111	111
a_4	000	000	000	111
a_5	000	000	000	000

当发行商分配码字时,它并不是简单地按顺序将 a_i 分配给第 i 个用户 u_i ,而是随机地选择一个置换 $\pi \in S_l$ (π 对用户是保密的),然后将 πa_i 作为第 i 个用户的指纹($\pi a_i = a_{i\pi(1)} a_{i\pi(2)} \dots a_{i\pi(l)}$),称上述编码方案为带有置换 π 的 $\Gamma_0(n,d)$ 编码方案。在文献[2]的定理 12 中作者指出:对于 $n \geq 3$,且 $\varepsilon > 0$,令 $d = 2n^2 \log(2n/\varepsilon)$,则带有 π 置换的 $\Gamma_0(n,d)$ 是带有错误概率 ε 的 n -安全 (l, n) 码。因为上述的码本 $\Gamma_0(n,d)$ 中码长为 $l = d(n-1)$,显然当数字产品的发行量较大时,该方案很不实用。

基于 $\Gamma_0(n,d)$,设用户数是 N ,文献[2]给出了一种码长正比于的 $\log^{O(1)} N$ 的 c -安全编码方案,其中 $c = O(\log N)$ 。其基本思想是用 n -安全 (l,n) 码的码本作为字母表。基本构造如下:

设 A 是 (L, N) 码, 其码元独立随机地取自尺寸为 n 的码表。将 $\Gamma_0(n, d)$ 作为该码表, 得到的新码本记为 $\Gamma'(L, N, n, d)$ 。设 $\tilde{a} \in A$, $\tilde{a}_1 \dots \tilde{a}_L$, 若 \tilde{a}_i 取原码表中第 t_i 个码字, 则新码本中相应码字的相应码元取自 $\Gamma_0(n, d)$ 中第 t_i 个码字。注意, 嵌入 L 个 $\Gamma_0(n, d)$ 中的码字时, 使用 L 个 S_L 中的置换。这 L 个置换对用户是保密的。此外, 码本 A 对用户也是保密的。文献[5]中的定理 17 指出了: 给定正整数 N, c 和 $\varepsilon > 0$, 令 $n = 2c$, $L = 2c \log(2N/\varepsilon)$, 且 $d = 2n^2 \log(4nL/\varepsilon)$ 则 $\Gamma'(L, N, n, d)$ 是带有错误概率 ε 的 c -安全码。该码本中有 N 个码字, 每个码字的长度为 $O(Ldn) = O(c^4 \log(N/\varepsilon) \log(1/\varepsilon))$ 。可见, 当 $c = O(\log N)$ 时, 码长为 $O(\log^{O(1)} N)$, 这大大增加了该方案的实用性。

尽管文献[2]中的编码方法具有较好的合谋容忍性, 但该文中的跟踪算法存在一个较大的问题, 即一个误发的随机错误就可能导致错误判断。文献[6]指出了若参与合谋的用户所对应的码字标号有较大间隔, 则会以较高概率产生误判。文献[11]将文献[2]中的假设予以放宽, 允许在不可侦察位以一定概率出现错误码元。文献[12]就 q 元码的情形对此问题进行了研究, 并提出了更切实际的嵌入假设。还有一些文献也对文献[2]中的编码方案进行了改进或补充。目前对这类编码方案的讨论主要集中于在一定的合谋容忍尺寸下, 如何降低用户的码长并尽量放宽嵌入假设, 以降低对嵌入算法强度的要求, 但值得注意的是, 改进跟踪算法的效率也是亟待解决的问题^[7]。

此外, 也可直接利用随机二进制码元作为指纹编码体制, 如 Lofvenberg 和 Wiberg 提出了一种二进制随机指纹编码方案^[13], 并给出了一种对可疑用户组进行检测的算法, 其码字的长度与抗合谋尺寸的指数函数成正比。这种方法编码思想较为简单, 但码字往往较长。

4.2.4 BIBD 编码

基于组合论的编码保证不同用户组合的合谋可行集均不同, 这样, 通过比较不同用户合谋后的可行集就可以追踪出合谋的用户。当数字指纹被正确提取出来以后, 将其与用户指纹集合比较, 如果发现其与某一指纹相同, 则可认为找到了非法再分发的用户。如果在指纹集合中没有发现与提取出的指纹相匹配的指纹, 则检查其是否在某一合谋可行集中。如果发现这一指纹属于某一合谋可行集, 则可通过该合谋可行集跟踪出合谋用户。

在文献[14]中, Trappe 等人提出了基于均衡不完全区组设计 (Balance Incomplete Block Design, BIBD) 构造的指纹编码方案, BIBD 码是组合数学中的一种编码方案, 它用来按一定的规则选择和安排事物。

组合设计是从一给定的集合中, 选定一组子集以满足某种特定的性质。指纹编码可以看成在一组位置中, 选取在其中的某些位置上置“1”, 其余的置“0”, 因而, 可以用组合设计来设计指纹编码。

设 $S = \{s_1, s_2, \dots, s_v\}$ 为一个有限集, $B = \{B_1, B_2, \dots, B_b\}$ 是 S 的子集的集合, 则 $\{S, B\}$ 称为某 S 上的一个设计。 S 中元素的总数称为设计 $\{S, B\}$ 的阶, 记为 $|S|$ 。

按照组合设计的传统定义方法, $\{S, B\}$ 中的子集称为区组。如果至少有一个区组中没有完全包括 S 中的所有 v 个元素, 则称该设计是不完全的。如果在一设计中, 各区组的容量 (即所包含的元素数) 相同, S 中各元素在 B 中出现的次数相同, 则称为区组设计。当任意一对元素 $s_i, s_j \in S (i \neq j)$ 在 B 中相遇的次数 λ_e 也相同, 则称该设计是均衡的。

设 $S = \{s_1, s_2, \dots, s_v\}$ 为包括 v 个不同元素的基集, $B = \{B_1, B_2, \dots, B_b\}$ 为 S 的 k -子集的

集合, r 为含有某任意一元素的 k -子集数, 且对任意一对元素 $i, j(i, j=1,2,\cdots, v, i \neq j)$, 有 λ_e 个区组同时包含它们, 则称 $\{S, B\}$ 构成的区组设计为均衡不完全区组设计, 简记为 $\text{BIBD}(v, b, r, k, \lambda_e)$ 。

例如, 区组设计

$$\begin{aligned} B_1: &1\ 4\ 7 & B_4: &1\ 5\ 9 & B_7: &1\ 6\ 8 & B_{10}: &1\ 2\ 3 \\ B_2: &2\ 5\ 8 & B_5: &2\ 6\ 7 & B_8: &2\ 4\ 9 & B_{11}: &4\ 5\ 6 \\ B_3: &3\ 6\ 9 & B_6: &3\ 4\ 8 & B_9: &3\ 5\ 7 & B_{12}: &7\ 8\ 9 \end{aligned}$$

在这个区组设计中, 元素数 $v=9$, 区组数 $b=12$, 每个元素出现次数 $r=4$, 每个区组包含的元素数 $k=3$, 任意一对元素相遇的次数 $\lambda_e=1$ 。根据定义, 例中的区组设计可表示为 $\text{BIBD}(9, 12, 4, 3, 1)$ 。

实际上, 所有 $\text{BIBD}(v, b, r, k, \lambda_e)$ 的各个区组, 都是从 v 个元素中任取 k 个的全组合区组中挑选出来的, 区组设计的五个基本参数 v, b, r, k, λ_e 之间存在着固定的关系: $bk = rv$ 和 $r(k-1) = \lambda_e(v-1)$ 。而 $\text{BIBD}(v, b, r, k, \lambda_e)$ 存在的必要条件是这两个关系必须成立。如果已知 v, b, r, k, λ_e 五个参数中的任意三个, 则其他两个参数也就随之决定了, 因此 $\text{BIBD}(v, b, r, k, \lambda_e)$ 也常简写为 $\text{BIBD}(v, k, \lambda_e)$ 。

在介绍抗合谋码之前我们先引入关联矩阵的概念。设 $B = \{B_1, B_2, \cdots, B_b\}$ 是 v 元基集 $S = \{s_1, s_2, \cdots, s_v\}$ 的一个 $\text{BIBD}(v, b, r, k, \lambda_e)$ 。关联矩阵 M 是一个 $b \times v$ 的 $(0, 1)$ -矩阵。

$$M = [m_{ij}], 1 \leq i \leq b, 1 \leq j \leq v$$

$$\text{其中 } m_{ij} = \begin{cases} 1, & s_j \in B_i \\ 0, & s_j \notin B_i \end{cases}$$

例如, 下面的 $\text{BIBD}(7, 3, 1)$ 矩阵 B 的关联矩阵可以表示为矩阵 M 的形式。

$$B = \begin{bmatrix} 1 & 2 & 4 \\ 1 & 3 & 6 \\ 1 & 5 & 7 \\ 2 & 3 & 5 \\ 2 & 6 & 7 \\ 3 & 4 & 7 \\ 4 & 5 & 6 \end{bmatrix}, \quad M = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

下面定义 $K-(G, \bullet)$ 抗合谋码的概念。

☒ 定义 4-1 $K-(G, \bullet)$ 抗合谋码。

设 G 为具有二进制运算 \bullet 的半群。 G^v 上的码 $C = \{c_1, \cdots, c_n\}$ 称为 $K-(G, \bullet)$ 抗合谋码或 $K-(G, \bullet)$ ACC, 如果满足所有 $0 < i \leq k, 0 < j \leq k, i \neq j$, C 中任取 i 个向量进行 \bullet 运算的结果与任取 j 个向量进行 \bullet 运算的结果不相同。当 $G = \{0, 1\}$ 且 \bullet 为逻辑与时, $K-(G, \bullet)$ ACC 可简称为 K AND-ACC。

这一定义与前面描述的抗合谋指纹思想本质上是一致的。根据抗合谋指纹思想, 可利用特殊的 BIBD 区组设计来构造指纹码字。定理 4-1 给出了 BIBD 与所需要的指纹码字之间的关系。

定理 4-1

设 $\{S, B\}$ 为 $\lambda=1$ 的 $(v, k, 1)$ -BIBD, M 为相应的关联矩阵。如果将 M 的行向量按位取补生成码向量, 则生成方案是 $k-1$ AND-ACC。

例如, 按上面的 BIBD $(7, 3, 1)$ 生成的码矩阵 C :

$$C = \overline{M} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

将多个用户按“逻辑与”合谋后, 产生的码字中的“1”称为特征码字, 不难看出, 特征码字为在合谋集的不可探测位置上标记为 1 的码字。因此, 可以根据合谋指纹 z 中的特征码字对合谋用户进行跟踪。例如, 将 C 的任意两个行向量进行“逻辑与”运算后, 得到的特征码字都是不相同的。

使用 BIBD $(v, k, 1)$ 设计生成 BIBD 抗合谋码, 用于数字指纹抗合谋攻击, 设计中的参数 b 对应于方案可容纳的用户数 n , 参数 v 为用户指纹的码长 l , $k-1$ 为方案允许的最大合谋用户数 c , 即在合谋用户数小于或等于 $k-1$ 时, 可跟踪到所有参与合谋的用户, 且 $n = (l^2 - l) / (c^2 + c)$ 。因而, 若该 BIBD 存在, n 用户的指纹只需要 $l = O(c\sqrt{n})$ 个基向量, 即指纹长度的近似随着 \sqrt{n} 线性增加, 随着 c 线性增加。相对于 C -安全码和正交码, BIBD 码能够极大地缩短指纹码字的长度。

在不扩展码字的情况下, C -安全码和正交码的码距为 1, 而 BIBD 码的码距相对要大一些, 即 BIBD 码的抗干扰能力比 C -安全码和正交码要强。另外, 可以对 BIBD 码的码字进行扩展, 使其能在“逻辑与”、“逻辑或”、“平均”和“随机选取”合谋方式下跟踪合谋用户。

BIBD 码用做指纹也存在问题, 即某参数下 BIBD 设计本身的存在性及相应的 BIBD 区组的获取都存在问题, 尤其指纹要求参数 $\lambda=1$ 。在参数比较大的情况下, 寻找 BIBD 区组的算法是相当复杂的。

其次, 对于 BIBD 合谋集的探测是通过特征码字进行跟踪的, 因而跟踪过程的匹配算法不是将合谋指纹与用户指纹进行直接匹配检测, 而是将合谋指纹与所有合谋用户数为 $c \leq k-1$ 的合谋集的特征码字进行匹配, 这样的合谋用户集数量 M 为

$$\begin{aligned} M &= \sum_{i=1}^{k-1} C_n^i \\ &= C_n^1 + C_n^2 + \cdots + C_n^{k-1} \\ &= \frac{n!}{1!(n-1)!} + \frac{n!}{2!(n-2)!} + \cdots + \frac{n!}{(k-1)!(n-k+1)!} \end{aligned}$$

当用户数 n 很大时或者 k 稍大时, 这一 M 就大得相当惊人了。在实际应用中, 则需要对 M 个用户合谋集的特征码字进行存储; 在跟踪检测时, 需要完成 M 次的匹配运算, 这一开销是无法接受的。

利用某些具有特殊组合性质的二进制（或多进制）码字对指纹编码进行研究，一直是指纹编码研究的热点之一。T. Lindkvist 在文献[15]中对一些特殊的二进制码进行了探讨。J. Domingo-Ferrer 等在文献[16]中提出了一种基于对偶二元汉明码的抗两个人合谋攻击的指纹编码方案。J. Dittmann 等在文献[17]提出了一种基于有限几何的指纹编码方案。Staddon, Stinson 和 Wei 在文献[18]中，对 IPP（Identifiable Parent Property，可确认父元）码、FP（Frameproof，防陷害）码、SFP（Secure FP，安全防陷害）码、TA（Tracibility，可跟踪）码的组合特性及它们的相互联系进行了研究；同时运用组合论和编码理论中的若干方法，讨论了有关码字结构中参数界的问题，并给出了几种关于 IPP 码、TA 码等的构造方法。

基于组合论的数字指纹有一些优点，比如，可以精确地确定多个合谋者；但是也存在一些问题，如当提取的指纹有损坏时，可能误将无罪用户检测为合谋者。这主要是因为，在提高数字指纹编码码距的同时，很难同时提高合谋可行集间的码距。在提取的指纹有损坏的情况下，有时很难判断提取的指纹属于哪个合谋可行集，错误的判断将导致冤枉无辜用户。基于组合论的数字指纹的思想还导致跟踪模式库比用户数庞大得多，用户数为 n 、最多容忍 N 个用户合谋的模式库大小至少为 $C_n^1 + C_n^2 + C_n^3 + \cdots + C_n^N$ ，跟踪算法效率较低^[19]。

4.2.5 基于残留特征跟踪的指纹编码

1. 基于残留特征跟踪的数字指纹抗合谋思想

针对基于组合论数字指纹所表现出来的这些不足，文献[20]提出了一种新的数字指纹抗合谋思想——基于残留特征跟踪的数字指纹思想，通过跟踪合谋残留的指纹特征来跟踪合谋者。

协同理论^[21]是 20 世纪 70 年代初由德国物理学家 Haken 提出并创立的一门新型学科。在协同模式识别领域，协同神经网络模型已被用于解决 2D 工业零件辨识、手写字符识别、人脸识别、车牌识别等问题。

在协同系统中，初始状态的设置表现为部分有序化的子系统，属于这个子系统的序参量在竞争中取胜，最后支配整个系统并使其进入这个特定的有序状态。换言之，一旦给出具有某个模式特征的集合，其中具有最强初始支撑的序参量就获得胜利，并驱使系统呈现模式中原来缺少的特征^[21]。

如果将多用户合谋产生的合谋模式作为协同系统的输入模式，而数字指纹库组成了协同系统的模式集合，则模式集合中具有最强初始支撑的序参量将获得胜利，该模式将被选择出来作为匹配的结果。也就是说，合谋指纹中具有特征多的对应用户将被认定为合谋者。

更简单地说，多个用户合谋时，谁残留的指纹特征更多，谁就将在协同神经网络的竞争中获胜，将被成功地跟踪。这就是基于残留特征跟踪的数字指纹抗合谋思想。

为了达到被合谋攻击时数字指纹的特征不被完全抹去，需要设计相应有效的特征编码。为方便表述，称参与合谋的指纹模式为合谋模式，合谋产生的模式为新模式，指纹库中模式本来就具有的特征称为原始特征，合谋产生的特征称为新特征。该编码必须满足两个条件^[20]：

- ① 合谋不能抹去合谋模式的所有原始特征，原始特征仍被尽可能多地保留下来；
 - ② 非合谋模式不能与新模式更匹配，即新模式尽可能少地具有指纹库中非合谋模式的特征。
- 实际上，大多数合谋是不会抹去原始特征的，只不过多个特征叠加起来形成了新特征。

如果模式库中没有非合谋模式与新模式更匹配, 协同系统会将新特征作为多个合谋模式的原始特征来对待。也可以从序参量的含义上来考虑这个问题: 若把输入模式看成原型模式的线性组合, 序参量就代表了输入模式对原型模式分解的系数, 输入模式越接近原型模式, 这个系数就越大, 相应的序参量就越大, 在竞争中获胜的可能性也就越大。也就是说, 如果不存在非合谋模式与新模式更匹配, 在协同系统中合谋产生的新模式就会主要分解在合谋模式上, 即新特征就会分解为多个合谋模式的原始特征。以上分析表明, 如果条件②得到满足, 条件①也就基本上得到了满足。

为满足上述编码要求, 提出线性无关特征码的概念^[20]:

对于 N 个向量 $R_i (i=1, 2, \dots, N)$, 这里, R 为二进制域向量, 如果

$$k_1 R_1 + k_2 R_2 + k_3 R_3 + \dots + k_N R_N = 0$$

仅当 $k_1 = k_2 = k_3 = \dots = k_N = 0$ 时才成立, 那么称这 N 个向量 $R_i (i=1, 2, \dots, N)$ 线性无关。

如果线性无关的 N 个向量 $R_i (i=1, 2, \dots, N)$ 满足

$$\max P(R_i, R_j) \leq \alpha, \quad i \neq j, \quad i, j = 1, 2, \dots, N$$

其中, $P(\cdot)$ 为某种相关性度量函数, α 为阈值, 那么, 称这 N 个向量为特征为 α 的线性无关特征向量。

给出另一种特殊的线性无关特征向量: 如果线性无关的 N 个向量 $R_i (i=1, 2, \dots, N)$ 之间的最小汉明距离满足

$$\min d(R_i, R_j) \geq \beta \cdot L, \quad i \neq j, \quad i, j = 1, 2, \dots, N$$

其中, β 为阈值, L 为向量 R_i 的长度, 那么, 称这 N 个向量为特征差为 β 的线性无关特征向量。

如果把线性无关特征向量作为数字指纹, 则把这种数字指纹编码称为线性无关特征码。

线性无关特征码保证了任意两个码向量之间都存在有效的区分度, 在一定程度上避免了数字指纹匹配时误匹配的情况发生。

线性无关特征码保证了任意多个码向量的线性组合都与其他码向量保持一定程度的差异, 这样就可以避免非法用户合谋陷害其他用户的可能性。同时, 线性无关特征码保证了参与合谋用户的数字指纹的一些特征不被完全抹去, 残留的特征多的数字指纹将在协同神经网络的竞争中获胜, 从而被成功地跟踪。

正交码是一种特殊的线性无关特征码。正交码不仅要求各个码向量之间是线性无关的, 而且要求各个码向量之间的内积为 0, 即

$$(R_i, R_j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad i, j = 1, 2, 3, \dots, N$$

正交码对指纹的提取要求较高, 它是靠提取的指纹全部特征来精确确定合谋用户的, 这必然导致数字指纹的鲁棒性不强。当有误码或者噪声干扰时, 有很大可能将无辜合法用户错误地判断为合谋者。

2. 基于残留特征跟踪的指纹编码

将用户信息编码为有意义的二值图像 (图 4-2)^[20]。
该图像大小为 64×15 , 阿拉伯数字为宋体小四号。

图 4-3 为基于残留特征跟踪的数字指纹方案^[20], 虚



图 4-2 数字指纹二值图像

线框中表示了系统中各个部分所采用的方法。方案主要由几个部分构成：数字指纹的生成方案、数字指纹的嵌入和提取方案、数字指纹的跟踪方案。

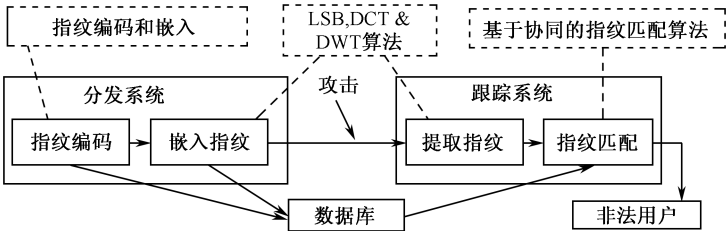


图 4-3 基于残留特征跟踪的数字指纹方案

指纹编码算法流程如下：

① 将用户的身份信息映射为一个数字序列号；

② 确定指纹生成密钥，该密钥决定指纹生成的一些参数，包括二值指纹图像的大小、阿拉伯数字的字体和大小、阿拉伯数字图像在整个二值指纹图像中的位置、各个数字图像之间的距离等；

③ 根据密钥生成数字指纹二值图像。

宋体阿拉伯数字 0~9 的图像矩阵如图 4-4 所示。

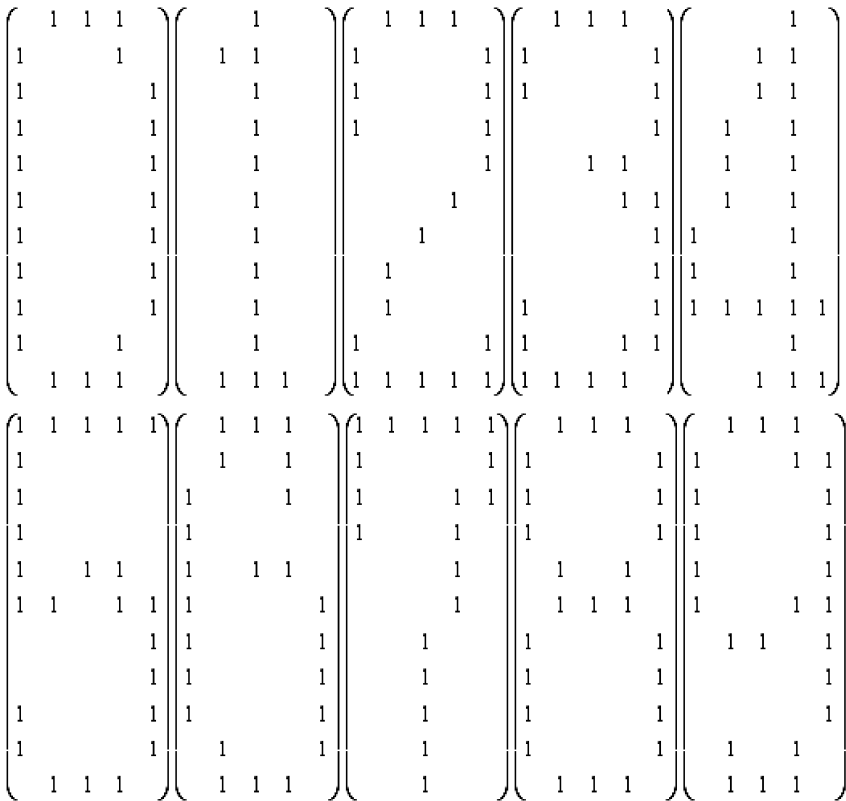


图 4-4 0~9 的图像矩阵

将每个矩阵的列分别合并为1列,得到10个向量。容易证明,这10个向量是线性无关的。因此,将这10个向量分别扩展,它们仍然是线性无关的。所以,按照上述算法生成的数字指纹库是线性无关的。文献[20]通过计算得出,构造的指纹库的特征差 $\beta=6/(15 \times 64)=0.00625$,即指纹库是由特征差为0.00625的线性无关特征码构成的。

这种编码扩展用户很方便,只要扩展数字的位数,每扩展一次,可以容纳的用户数都增加为扩展之前的10倍,而码长仅增加1个阿拉伯数字图像向量的长度。一个阿拉伯数字图像向量的大小取决于阿拉伯数字的字体和大小。文献[20]中的阿拉伯数字字体均为小四号宋体,每个阿拉伯数字编码后数据长度为 $11 \times 5=55$ 位。

数字指纹跟踪过程就是在跟踪模式库中找到与所提取的指纹相匹配的模式,并确定该指纹模式对应的用户信息的过程。文献[20]方案中采用基于协同学的数字指纹匹配算法,跟踪模式库就是用户指纹库。

数字指纹模式匹配如图4-5所示。

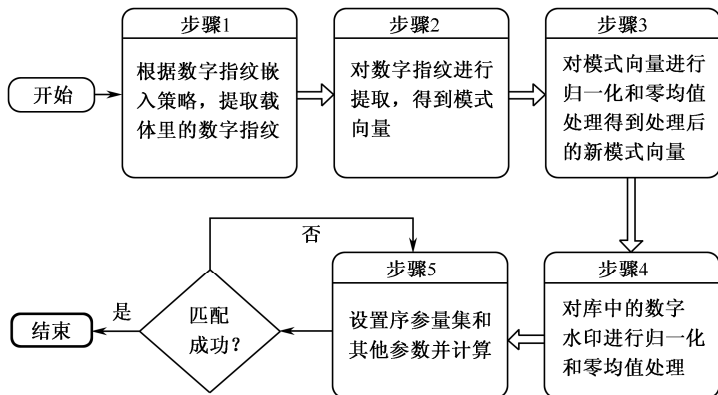


图4-5 基于残留特征跟踪的数字指纹模式匹配流程

匹配步骤:

- ① 根据数字指纹嵌入策略提取载体里的数字指纹。
- ② 对该数字指纹进行模式提取,可以得到模式向量。
- ③ 对该模式向量进行归一化和零均值处理,得到处理后的新模式向量。
- ④ 对数字水印库分别进行归一化和零均值处理,得到原型模式向量集和伴随向量。同时求出初始序参量集,并将参数代入协同演化模型进行序参量演化,可以得到新的序参量集。
- ⑤ 如果序参量中有一个为1,则待匹配的数字指纹为指纹库中该序参量所对应的那个指纹,结束匹配;否则,继续匹配。

在保证正确跟踪的成功率的同时,该指纹方案与基于组合论的数字指纹方案相比,在系统性能方面具有以下优点^[20]。

(1) 编码的效率

数字指纹可以为有意义的图像,方便了用户和发行商。固有的数字指纹方案都靠编码来保证抗合谋特性,将用户信息编码为无意义的0和1序列。有意义的数字指纹可以方便发行商追踪非法用户,在数字指纹损坏程度不大的情况下,可以直接辨认数字指纹,省去了数字

指纹的匹配和跟踪过程。同时，在文献[20]中，码长每增加 55 位，可容纳的用户数就增加 10 倍，编码的效率比基于组合论的编码都要高得多。

(2) 跟踪算法效率高

当前，基于组合论编码的数字指纹方案的跟踪算法要求在大小至少为 $C_n^1 + C_n^2 + C_n^3 + \dots + C_n^N$ (n 为用户数， N 为抵抗合谋攻击的最大合谋用户数) 的模式库中运行匹配算法，而基于协同学的数字指纹跟踪方案^[20]中，模式库大小只需要为 n 。

(3) 至少可以成功跟踪到 1 个合谋者

序参量演化曲线可以作为参考来确定其他合谋嫌疑人。被成功跟踪的数字指纹序参量最后演化为 1，而与该曲线最靠近的序参量代表的用户可能是合谋的参与者。

由于要对抗合谋攻击，需要对指纹进行一些特殊的编码，同时对要嵌入信息的长度也提出了更高的要求。如何结合嵌入技术的发展状况设计较短的实用的码字，仍将是指纹编码中的核心课题^[7]。

4.3 数字指纹协议

4.3.1 对称数字指纹协议

对称数字指纹协议是由 B. Chor 等人于 1994 年提出的^[5]。这种方案中指纹的嵌入是由发行商来完成的，因为发行商知道所嵌入的指纹，所以发生盗版时，发行商不能向第三方出示有力的证据证明盗版的来源一定是授权的原始购买者，即对称指纹方案不能提供不可否认性。

在详细描述这个协议之前举一个典型的应用例子。在付费电视广播系统中，发行商向他的所有合法用户广播加密的电视节目。为了使合法用户能够解密所有节目块，发行商给每个合法用户分发一个唯一的个人密钥。用户使用一个有效的个人密钥来获取节目的内容，该个人密钥就可以理解为指纹协议中的“指纹”。

这里有几个假设，任何用户都需要使用一个有效的个人密钥来获取节目的内容，而不能把节目的明文广播给非法的用户；发行商不可能用每个合法用户的个人密钥直接加密节目，或者说向每个合法用户广播完全不同的加密节目内容，因为这样广播的代价太大而不可行；如果节目使用同一密钥加密，则一个合法用户只要用他的个人密钥解密出会话密钥并发给一个非法用户，后者就可以一直解密所有的节目。所以节目必须被分成许许多多的小块，每个块使用不同的对称密钥加密，盗版者只能把自己的个人密钥发给非法用户使用，或几个合谋者通过比较他们各自的个人密钥生成一个新的个人密钥。

在后面的描述中将说明对称指纹协议在盗版者合谋的情况下，发行商也可以追查出至少一个合谋者，即叛逆者追踪。更一般的情况下，叛逆者追踪描述成如下协议模型。

1. 协议中的对象

① 产品：发行商发布的一个数字产品，如一部电影。合法用户的个人密钥在同一个商品中都是有效的，每个商品被分成多个会话数据块。

② 会话密钥：对称密码体制的密钥。

③ 会话数据块：发行商向外发布的一段产品数据，每个会话数据块包含两部分。

密文块：由会话密钥加密的有效数据，每个会话数据块中使用的会话密钥都不相同。

使能块：用分段并且对称加密的方法保存对应加密块使用的会话密钥。

④ 个人密钥：每个合法用户用来解密会话密钥的一个密钥集合。

⑤ 码字：和个人密钥对应的唯一标识一个合法用户的长为 L 的代码。

2. 协议参数

coll_size：协议可以保证安全的最大的合谋者的个数。

L ：会话密钥的分段个数，个人密钥的集合大小。

b ：一个任意字母表的大小，如字母表为 $\{1, 2, 3, \dots, b\}$ 。

N ：购买者的最大数量。

3. 协议过程描述

对称指纹协议可以简略地描述成以下过程（图 4-6）。

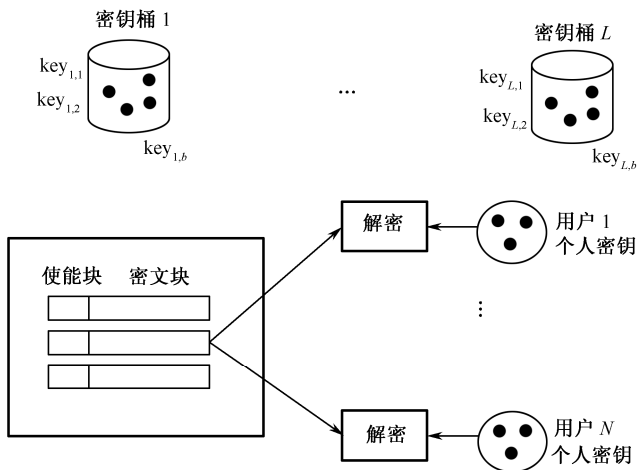


图 4-6 对称数字指纹协议的描述

（1）密钥初始化子协议

该协议由发行商执行。对于一个产品，发行商随机选择 L 个对称密钥集合，每个集合有 b 个密钥，每个集合可以形象地比做一个密钥桶，这 $L*b$ 个对称密钥可以按顺序表示为 $key_{L,b}$ 。对于这个产品中每个会话数据块，选择一个会话密钥 S 。

（2）密钥指纹子协议

对每个购买者，发行商统一随机选择一个长为 L 的码字，其中每一位都是字母表 $\{1, 2, 3, \dots, b\}$ 中的某个字母。这样一个码字就和 L 个密钥桶中 L 个密钥对应起来，这 L 个密钥构成一个购买者的个人密钥。例如，码字为 $codeword = (7, 3, 4, \dots, 9)$ ，则个人密钥 $key = \{key_{1,7}, key_{2,3}, key_{3,4}, \dots, key_{L,9}\}$ 。发行商保存每个购买者的个人密钥，并把个人密钥 key 发给对应的购买者，个人密钥对其他人是保密的。码字的可能空间为 $b^L \gg N$ 。

(3) 会话发送子协议

发行商广播（发布）产品中每个会话数据块，每个会话数据块中密文块使用这个块对应的会话密钥加密。会话密钥被等长的分成 L 段 $S_1, \dots, S_i, \dots, S_L$ ，其中第 i 段使用第 i 个密钥桶中的 b 个密钥进行加密（ $i=1, \dots, L$ ）。这 $L*b$ 个加密值按次序排列构成这个会话数据块的使能块，随加密块一起发送。每个购买者收到一个会话数据块后，使用自己的个人密钥从使能块中解密出对应的 S_1, \dots, S_L ，组成完整的会话密钥 S ，从而解密出对应加密块中的有效数据。

(4) 盗版跟踪子协议

对一组 coll_size 个合谋叛逆者来说，为了使一个盗版用户可以成功解密所有会话数据块中的数据，他们必须提供一个有效的个人密钥。首先他们不会提供其中单个人的个人密钥，否则发行商可直接从检测出的这个人密钥查到叛逆者，那么他们就要对 L 段中每个 S_i ，从他们的 coll_size 个个人密钥中对应的 coll_size 个 $\text{key}_{i,j}$ 中选择一个作为解密 S_i 的密钥，使最终共谋产生的个人密钥是有效的，但是和叛逆者中的任意一个个人密钥都不同，以阻止追查。但是，该协议通过如下算法使发行商仍能确定其中的一个叛逆者：当事后查到了一个盗版用户，发行商可以得到一个该盗版用户使用的个人密钥 K （实际是每个密钥桶中对应一个 $\text{key}_{i,j}$ ）。对于这个 K 中 L 个 $\text{key}_{i,j}$ （ $i=\{1, \dots, L\}$ ），发行商对每个其个人密钥中对应位置等于 $\text{key}_{i,j}$ 的购买者做一次标记，如 $K=\{\text{key}_{1,7}, \text{key}_{2,3}, \text{key}_{3,4}, \dots, \text{key}_{L,9}\}$ ，某个购买者的个人密钥 $\text{key}=\{\text{key}_{1,7}, \text{key}_{2,8}, \text{key}_{3,4}, \dots, \text{key}_{L,6}\}$ （假设各个位置中的对应 key 都不相同），则这个购买者被标记两次（第 1、3 两个密钥桶）。最后被标记次数最多的购买者就被认为是叛逆者之一。实际上这个购买者至少被标记 $L/\text{coll_size}$ 次，并且合谋所伪造的个人密钥等于一个合法购买者的个人密钥的概率根据参数设置是可忽略的，以保证共谋不能陷害一个诚实的购买者，这个算法的证明以及合谋不可陷害的证明参见文献[5]。

4.3.2 非对称指纹协议

1. 非对称指纹协议的基本思想

在早期的指纹方案中，通常由发行商生成指纹复制品发放给用户。发行商和购买者都知道卖给购买者的产品的指纹复制品（如，个人密钥），那么在发现盗版提起诉讼时，购买者完全可以声称发行商发现的指纹拷贝是发行商的一个不诚实的员工所流传出去的，从而使仲裁无法确认谁是叛逆者（所谓对称性）。针对这一问题，1996 年 Pfitzmann 和 Schunter^[23]引入了非对称指纹的概念，并应用于叛逆者追踪（Traitor-tracing）协议，以改善协议的安全性和公平性。

非对称的概念就是要使标识购买者的“指纹”只有购买者自己知道，而发行商在商品交易时不知道，但在追踪时发行商可以得到足够的证据来指认相关的叛逆者。类似于非对称的加密体制能够实现不可否认性，非对称指纹体制最主要的特点是实现非法用户的不可否认性。非对称数字指纹有以下几个方面的含义^[23]：

① 只要协议正常执行, 用户可以得到含有其指纹的合法复制品;

② 对发行商而言, 在一定的合谋尺寸下, 发行商能够从非法复制品中跟踪出至少一个非法分发者, 同时能够提供证明用户有罪的证据 (该证据是不可伪造的);

③ 对用户而言, 无论合谋人数的多少, 无辜用户不能受到陷害。

非对称指纹体制一般由 4 个基本协议组成: 初始化协议 (用户进行购买登记和发行商的有关初始化工作)、指纹添加协议 (为用户生成带指纹的复制品)、跟踪协议 (确认非法分发者的身份)、审判协议 (发行商向第三方提供用户有罪的证据)。目前, 非对称指纹体制的构造手段主要有基于一般的安全多方计算协议 (如文献[24])、利用特殊的密码学协议 (如文献[25])、利用密码算法 (如文献[26]) 等。安全多方计算是自然的实现思路, 实际也是基于密码协议的研究。因为对其有效实现的研究不充分, 人们便寻求能够避免使用一般的安全多方计算协议的设计方法。以特殊的、更为具体的密码协议为基础和直接应用密码算法便是两种典型的思路。此外将公钥密码算法与防篡改硬件相结合也是一种设计非对称数字指纹体制的思路。文献[27, 28]中就用到了防篡改硬件。这种方法的特点是设计思想简单, 但这种体制的安全性不仅基于密码算法的安全性, 而且还基于硬件的安全性。尽管从实际应用角度看实现效率较高, 但同时增加了硬件开销。因此, 如何利用特殊的密码协议或者直接利用密码算法的有关研究成果, 设计有效实用的非对称指纹体制是非常有意义的应用基础研究工作。

值得注意的是, 非对称指纹协议的设计与指纹的编码是密切相关的。指纹编码所讨论的主要问题是使指纹体制能够抵抗合谋攻击。就目前来看, 已经有文章讨论抗合谋攻击能力较强的非对称指纹体制的设计, 但多基于安全多方计算^{[24][29]}, 而不基于一般安全多方计算的非对称指纹体制^{[25][26]}, 并没有对抗合谋攻击能力做出具体分析。如何设计既有较好抗合谋攻击能力, 又有较好实现效率的非对称指纹方案; 或在已有的具有较好抗合谋攻击能力的指纹编码方案的基础上, 设计能够有效实现的非对称指纹协议, 都是值得进一步研究的方向^[7]。

2. 基于非对称指纹的叛逆者追踪协议

下面简单介绍 Pfitzmann 等设计的基于非对称指纹的叛逆者追踪协议^[23]。该协议参与的角色和对象与 4.3.1 节的对称型叛逆者追踪协议一样, 需要增加三个密码学体系: 一个公钥签名模式、一个承诺模式和一个安全两方计算协议。

(1) 协议参数

σ : 一个适当的概率参数以表示安全的系数。

coll_size : 协议可以保证安全的最大的合谋者的个数。

L : 会话密钥的分段个数, 个人密钥的集合大小, 这里选定

$$L = 64 * \text{coll_size} * (\sigma + \log_2(N))$$

b : 一个任意字母表的大小, 如字母表为 $\{1, 2, 3, \dots, b\}$, 这里选定

$$b = 48 * \text{coll_size}$$

N : 购买者的最大数量。

(2) 协议过程

① 购买者的签名密钥生成: 每个购买者生成一对签名模式使用的公钥 PK_B 和私钥 SK_B , 并公布其公钥 PK_B 。

② 密钥初始化子协议：和对称指纹协议中的对应子协议相同。购买者选择 L 个密钥桶，每个桶有 b 个对称密钥。

③ 密钥指纹子协议：购买者选择一个长为 L 的码字 word_B ，对 word_B 生成一个承诺 com_B ，并把这个承诺发送给发行商。用来揭示承诺的信息为 open_B 。

购买者使用自己的私钥 SK_B 对如下消息进行签名： $\text{msg}_B = (\text{text}, \text{com}_B)$ ，生成 sig_B 。其中 text 是一个字符串，说明签名消息的含义。发行商使用 PK_B 验证 sig_B 是对 msg_B 的一个有效签名。接下来执行安全两方计算协议。

输入如下。

购买者： word_B 和 open_B 。

发行商：在密钥初始化子协议中生成 $L*b$ 个对称密钥，一个随机选择的大小为 $L/2$ 的 $\{1, \dots, L\}$ 的子集 Set_B ， com_B 。

计算如下。

验证 open_B 可以打开承诺 com_B ，否则协议失败。

验证 Set_B 的大小最多为 $L/2$ 个元素。

word_B 中符号对应集合 Set_B 中元素的位置组成的有序符号集称为 halfword_trace_B ，其余部分称为 halfword_evid_B 。

输出如下。

购买者：由 word_B 对应生成的个人密钥。

发行商： halfword_trace_B ，发行商只知道 word_B 中的一半符号而不能猜出整个 word_B ，符合非对称性。

执行安全两方计算后，发行商得到的交易记录 $\text{record}_M = (\text{id}_B, \text{text}, \text{com}_B, \text{sig}_B, \text{Set}_B, \text{halfword_trace}_B)$ ，购买者得到的交易记录 $\text{record}_B = (\text{text}, \text{word}_B, \text{open}_B)$ 。

④ 会话发送子协议。

发行商发布每个会话数据块，会话密钥被分成等长的 L 段 $S_1, \dots, S_i, \dots, S_L$ ，其中第 i 段使用第 i 个密钥桶中的 b 个密钥进行加密 ($i=1, \dots, L$)。这 $L*b$ 个加密值按次序排列构成这个会话数据块的使能块，随密文块一起发送。每个购买者收到一个会话数据块后，使用 word_B 得到个人密钥，并用个人密钥从使能块中解密出对应的 $S_1, \dots, S_i, \dots, S_L$ ，组成完整的会话密钥 S ，从而解密出对应加密块中的有效数据。

⑤ 盗版跟踪子协议。

发行商发现一个盗版用户，得到一个个人密钥，也就相当于找到一个长为 L 的码字 $\text{word}_{\text{found}}$ ，搜索所有交易记录中的 $(\text{Set}_B, \text{halfword_trace}_B)$ ，找到一个记录，其 halfword_trace_B 和 $\text{word}_{\text{found}}$ 至少有 $L/(4*\text{coll_size})$ 个对应位置上符号是相同的。他取出 $\text{msg}_B = (\text{text}, \text{com}_B)$ 和 sig_B 准备向仲裁者控告此购买者。

⑥ 仲裁子协议。

发行商提供的证据为 $\text{proof} = (\text{msg}_B, \text{sig}_B, \text{word}_{\text{accuse}})$ ，其中 $\text{word}_{\text{accuse}}$ 为长为 L 的码字，码字中对应 Set_B 中元素的位置上由 halfword_trace_B 中的对应符号组成，码字中其他位置的符号由 $\text{word}_{\text{found}}$ 中的对应位置符号组成。

购买者提供 $open_B$ 。

仲裁者首先验证 sig_B 签名的有效性和承诺的有效性，揭示 $word_B$ 。然后验证 $word_{found}$ 和 $word_B$ 是否至少在 $L/2+L/(16*coll_size)$ 个对应位置上的符号相同，成立则指控成功，否则购买者即可否认指控。

为了使 $word_B$ 始终对发行商保密，可以用零知识证明来给出 $word_B$ 。非对称指纹协议可以简略地描述成如图 4-7 所示的过程。

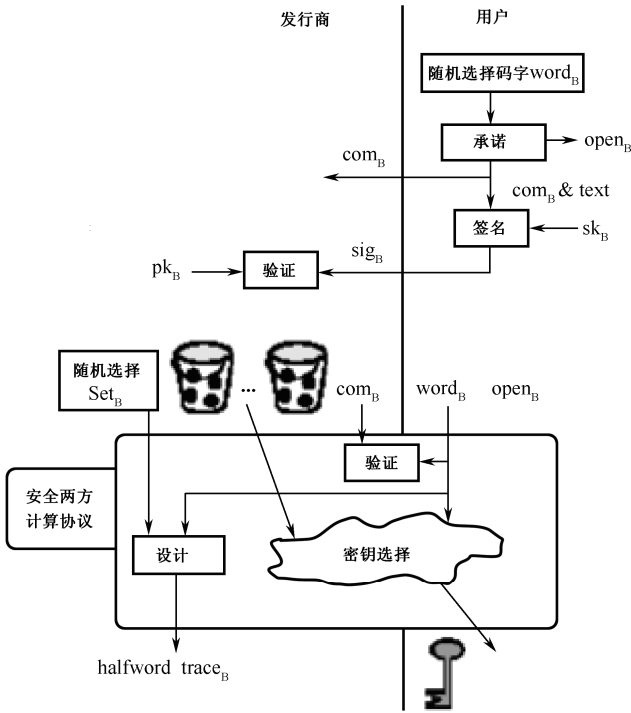


图 4-7 非对称指纹协议描述

3. 对购买者透明的非对称扩频指纹方案

非对称数字指纹的概念一经提出便引起了研究者的广泛关注。文献[30]中 Memon 和 Wong 提出了一种不使用安全多方计算协议的非对称指纹体制，并将扩频技术引入了指纹方案，提出了买卖双方（Buyer-Seller）水印协议，为数字指纹实用化提供了新思路。但该体制需对要发行复制品进行公钥密码体制中的加密运算。众所周知，公钥密码算法相对于对称密码算法效率低得多，当要发行的复制品数据量较大时，该方法的效率将很低^[31]。

文献[32]在直接序列扩频编码基础上，提出了一种对购买者透明的非对称指纹方案。该方案包括以下实体：发行商（M）、购买者（B）、指纹中心（FC）、认证中心（CA）和仲裁者（A）。认证中心为协议各方分配密钥，指纹中心在协议构造指纹并存储有关指纹信息，并且假定认证中心与指纹中心是可信第三方（TTP）。

方案中还使用了一个具有加法同态性质的密码系统。因此我们在介绍算法前先了解扩频和同态的概念。

定义 4-2 扩频指纹码^[32]

根据直接序列扩频编码的性质，令 Σ 表示正态分布的实数空间，设集合 $\Gamma = \{W^{(1)}, W^{(2)}, \dots, W^{(n)}\} \subseteq \Sigma^l$ 称为一个扩频指纹码，如果 Γ 中的每个码字 $W^{(k)} = (w_1^{(k)}, w_2^{(k)}, \dots, w_l^{(k)}) (k=1, 2, \dots, n)$ 是相互正交的随机序列，其中 $w_i^{(k)} (i=1, 2, \dots, l)$ 是根据正态分布 $N(0,1)$ 独立选取的，则称 Γ 中所有码字构成的集合为 Γ 的码书。

令 $X = (x_1, x_2, \dots, x_l)$ 表示需要嵌入指纹的作品，信号 x_i 由感知模型计算求得的指纹权重因子为 g_i ，则扩频序列 $W^{(k)}$ 的嵌入公式为

$$x_i^{(k)} = x_i + g_i w_i^{(k)}, \quad i=1, 2, \dots, l$$

嵌入后作品的带指纹复制品为 $X^{(k)} = (x_1^{(k)}, x_2^{(k)}, \dots, x_l^{(k)})$ 。

定义 4-3 同态^[33]

对于两个代数结构 A 和 B ，其中 \circ 是 A 中的运算， $*$ 是 B 中的运算， $\forall x, y \in A$ ，有 $f(x \circ y) = f(x) * f(y)$ ，则映射 $f: A \mapsto B$ 称为 A 到 B 的同态。

对于公钥加密算法 $E(\cdot)$ ，如果给定 $E(x)$ 和 $E(y)$ ，在没有私钥的情况下能够计算出 $E(x \circ y)$ ，则称该公钥加密算法具有同态性质。例如，RSA 公钥密码算法具有乘同态性质，而 Paillier 算法具有加同态性质。

方案包括四部分协议：初始化协议、购买协议、叛逆识别协议和审判协议。

(1) 初始化协议

认证中心选择一个具有加法同态性质的密码系统，为每个参与方分配私钥和公钥构成的密钥对 (sk, pk) ，同时 CA 选择一个强碰撞自由的 Hash 函数，函数输出长度为密码系统明文长度的一半，CA 将此函数公开。此外，协议各方将其具有的公钥公开。

(2) 购买协议

购买协议是购买者 B、分发商 M 和指纹中心 FC 之间的三方协议。协议的输入是 B 的公钥 pk_B 秘密信息 text，以及 FC 的指纹序列 W ；协议执行后的输出是 B 获得带指纹的复制品 X_B'' ，M 得到销售记录 Record，FC 存储提供的指纹序列 W_B 和签名 $\text{sign}_B(W_B)$ 。由扩频指纹和同态密码构造购买协议。

(3) 叛逆者识别协议

设发行商 M 得到一份非法复制品 X ，M 使用指纹提取算法在指纹 V 的嵌入位置得到指纹 V ，并译码得到其中的记录号 No_M ，或者通过与销售记录中指纹 V 的相关检测查找到记录号 No_M ；在合谋攻击情况下，可能跟踪到由多个叛逆者组成的集合。如果能够寻找到销售记录，由记录则可确定出叛逆者的身份 ID_B 。

(4) 审判协议

审判协议是仲裁者 A、购买者 B、发行商 M 和指纹中心 FC 参加的 4 方协议。对非法作品审判可以在没有购买者 B 参加的情况下进行，并且对于仲裁结果，购买者 B 可以通过公布自己的秘密信息进行辩护，同时审判不会危及系统的安全。

协议的详细过程请参阅文献[32]。算法在扩频指纹编码基础上，根据扩频指纹，并结合加

法同态密码系统,给出了一种对购买者透明的非对称指纹方案,实现了透明购买、不可感知、缺席审判等特性。与其他指纹方案相比更具实用性和安全性,为实现公平电子交易以及数字指纹提供了一条新思路。

4.3.3 匿名指纹

1. 匿名指纹的基本思想

无论是对称还是非对称指纹协议中,用户均须在购买过程中提交自己的身份信息,这破坏了购买过程的隐秘性。正是在这种背景下,在提出了非对称指纹协议后,1997年 B.Pfitzmann 和 M.Waidner 在文献[34]中首次提出了匿名数字指纹协议的概念,其目的是保持购买者在进行交易时能不对发行商泄露其真实身份。但是,在发现产品被用户非法分发时发行商仍能检测出叛逆者的身份。匿名指纹类似于盲签名,它使用一个可信的称为注册中心的第三方来识别被怀疑有非法行为的用户。发行商若没有注册中心的帮助就不能识别他。通过使用注册中心,发行者不再需要保存用户和指纹的相对应的详细记录。

一个匿名指纹体制应满足以下要求。

- ① 正确性:只要协议的各个部分都能够成功执行,最后用户将得到其要购买的拷贝。
- ② 无辜用户的安全性:无辜的用户不会受到陷害。
- ③ 发行商的安全性:当发现非法拷贝时,发行商能够凭借其中的指纹信息对用户进行跟踪。
- ④ 匿名性和不可连接性:如果没有拿到用户非法分发的拷贝,发行商即使与登记中心进行联合也不能确定用户的身份,而且发行商即使与登记中心进行联合也不能将同一个用户的不同购买进行联系。

B. Pfitzmann 在文献[34]给出了实现匿名数字指纹体制的模型,并讨论了几种变形;同时给出了在某些假设下实现匿名指纹的一种框架。匿名指纹协议的实现通常是引入一个注册中心,负责为用户的真实身份进行登记,购买者选择一个假名,即签名方案中的一个密钥对 (sk_B, pk_B) ,同时从注册中心获得一个证书 $cert_B$,有了这个证书注册中心就知道了这个假名的购买者的真实身份。当购买者购买商品时,他用标识这次购买的文本(text)计算出签名 $Sig = \text{sign}(sk_B, \text{text})$,然后将信息 $Emb = (\text{text}, Sig, pk_B, cert_B)$ 嵌入购买的数据中。购买者在比特承诺中隐藏这个值,并以零知识方式向发行商发送证书和承诺。当需要鉴别时,发行商提取出 Emb 并给注册中心发送 $\text{proof} = (\text{text}, sig, pk_B)$,并要求验证。作为回答,注册中心向发行商发回购买者的签名,发行商可以用这个签名来验证所有的值并且指控购买者。但是 B.Pfitzmann 所提出的匿名指纹框架^[34]过于依赖零知识证明,从而不具备实用价值。

2. 基于群签名的匿名指纹方案

近年来,人们对匿名指纹方案进行了大量的研究。Pfitzmann 和 Sadeghi 在文献[35]中提出了一个高效的匿名指纹方案,但该方案有两个缺点:

- ① 用户进行每次交易前都必须在注册中心注册一次;
- ② 发行商必须与注册中心合作才能恢复出非法用户的身份。

为了克服这两个缺点，Camenisch^[36]提出了利用群签名来构造匿名指纹的框架。

群签名是一种具有可撤销匿名性的数字签名技术，基于群签名的匿名指纹方案利用了群签名的匿名性和可跟踪性，其匿名性可为合法用户提供匿名保护，其跟踪性又使得可信机构可以跟踪违法行为。匿名指纹方案中的可信任第三方 RC（注册中心）扮演群签名方案中的群管理员；每个购买者在 RC 注册后就成为了群中的群用户，这个群包含所有注册的购买者。购买者以整个群的名义对描述该次交易的文本进行签名，该签名就如同一个证书。发行商验证签名合法后，获得了撤销管理员的公钥。一旦发现非法拷贝，发行商提取出其中嵌入的秘密信息撤销管理员的私钥，就可以利用群签名的性质公开进行非法拷贝的购买者的身份。

J. Camenisch 的基于群签名匿名指纹方案^[36]的主要思想（图 4-8）包括以下四个协议。

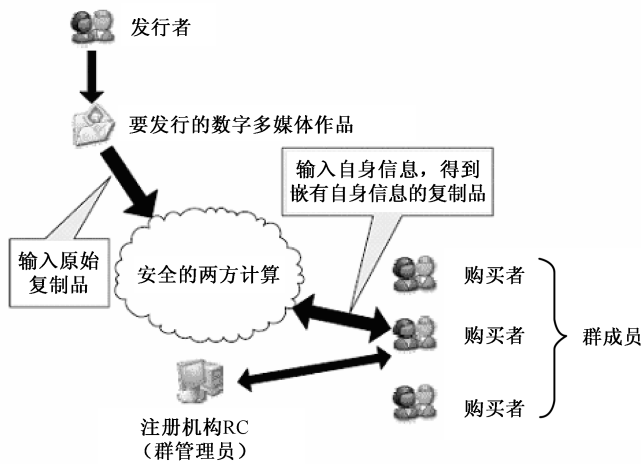


图 4-8 J. Camenisch匿名指纹方案

- (1) 用户注册协议
- 注册中心 RC 相当于群签名方案中的群管理员，通过群签名中密钥生成协议 GKG-M 得到密钥对 (x_M, y_M) ，并公开 y_M ，购买者 B 和群管理员 RC 通过 G-Join 获得密钥对 (x_B, y_B) ，注册中心则保留记录 (ID_B, y_B) 。
- (2) 指纹嵌入协议
- 购买者 B 用群签名中的密钥生成协议 GKG-R 获得密钥对 (x_R, y_R) ，然后用群签名方案对消息 m 进行签名，得到签名 $\delta = \text{Sig}(x_B, (y_R, y_M), m)$ 。B 将签名 δ 、 y_R 和 $\text{com}(x_R)$ 发送给发行商 M，并向 M 证明 $\text{com}(x_R)$ 确实是 y_R 对应的私钥 x_R 的承诺。
- 发行商 M 通过群签名方案中的验证协议 G-Ver 验证 δ 是否是一个合法的群用户签名。如果签名合法，M 保存记录 (y_R, δ) 。
- 在安全的通信环境下，发行商 M 和购买者 B 双方进行两方安全计算嵌入协议 Emb。其中 M 输入原始复制品 P_0 和 $\text{com}(x_R)$ ，而 B 输入 $\text{com}(x_R)$ 和 x_R ，结果 B 得到一个 P_0 的合法复制品 P_B 。
- (3) 身份识别协议
- 如果发行商 M 发现某个非法复制品，并从其中提取出购买者 B 的秘密信息 x_R ，就可以计算得到 y_R ，并在数据库找到相应的记录 (y_R, δ) ，利用群签名中的追踪算法

$G\text{-Open}(x_R, (y_R, y_M), m, \delta))$ ，发行商能够得到该签名者即非法购买者的身份。

(4) 仲裁协议

发行商将相关的证据发给仲裁者 A，A 通过验证 M 提供的证据，给出最后的仲裁结果。该方案主要是利用群签名的性质来隐藏用户的身份，但是存在两个主要缺点^[37]：

① 嵌入的过程采用了两方安全计算，使得该方案并不实用，J. Camenisch 在文献[36]中也承认该方案在目前并不可行。

② 该方案是准匿名的，如果注册中心 RC 和发行商 M 联合，则用户的身份不能隐藏。

3. 基于加同态公钥密码体制的匿名数字指纹方案

自从 Pfitzmann 和 Waidner 在文献[34]中引入了匿名指纹的概念以来，已有许多匿名指纹方案被提出。但是，大多数匿名指纹方案由于基于过于复杂的密码协议而在实际应用中并不可行。因此，如何避免使用复杂协议构造匿名数字指纹方案是数字指纹研究需要解决的一个关键问题^[33]。

在文献[30]中，Memon 和 Wong 利用公钥密码算法的同态性质提出了一种数字多媒体作品的买卖协议；但是，该方案不具备为购买者提供匿名购买的能力，而且在发现非法分发的数字作品时，销售商需要被指控的购买者参与并提供自己的秘密信息才能解决版权纠纷的问题。尽管利用公钥密码算法的同态性质构造匿名指纹方案因其简单实用受到研究者的广泛关注，也取得了一定的研究成果，但是，匿名指纹方案与密码协议以及密码算法密切相关，到底如何去构造具有匿名功能的公钥和私钥对，并且保证指纹嵌入的非对称性，仍然是基于同态公钥密码算法的匿名指纹技术没有解决的一个瓶颈问题。

文献[33]提出了一种基于加同态公钥密码算法的匿名数字指纹方案，在发现盗版的情况下，发行商不需要第三方的帮助就能鉴别出数字多媒体作品的非法分发者，解决版权纠纷时也不需要购买者参与并提供相关的秘密信息，从而达到实现两方审判的目的。该方案具有用户匿名及不可关联、销售商的可保证安全性和用户的可保证安全性等特点。

为了构造一种不需要第三方的帮助就能鉴别出盗版者的匿名指纹方案，采用 Bresson 等人提出的公钥密码算法^[38]，算法描述如下。

(1) 参数设置

设 $N = pq$ ，其中 p 和 q 为素数，且 $p = 2p_0 + 1$ ， $q = 2q_0 + 1$ ，而 p_0 和 q_0 也为素数， G 为模 N^2 的二次剩余循环群。

(2) 密钥生成

随机选择 $\alpha \in Z_{N^2}^*$ 和 $a \in [1, \text{ord}(G)]$ ，并使 $g = \alpha^2 \bmod N^2$ ， $h = g^a \bmod N^2$ ，那么公钥为 (N, g, h) ，对应的私钥为 a 。

(3) 加密

对于明文 $m \in Z_N$ ，在 Z_{N^2} 中选择随机数 r ，按下列方式计算密文对 (A, B) ：

$$A = g^r \bmod N^2, \quad B = h^r (1 + mN) \bmod N^2$$

(4) 解密

有两种解密方式, 其中一种解密方法是已知密钥 a , 按下面的公式计算明文:

$$m = \frac{B / A^a - 1 \bmod N^2}{N}$$

对于明文 m_1 和 m_2 , 如果使用 Bresson 密码算法对它们进行加密, 那么其密文分别为 $E(m_1) = (A_1, B_1)$ 和 $E(m_2) = (A_2, B_2)$, 其中:

$$\begin{aligned} A_1 &= g^{\eta_1} \bmod N^2, & B_1 &= h^{\eta_1} (1 + m_1 N) \bmod N^2 \\ A_2 &= g^{\eta_2} \bmod N^2, & B_2 &= h^{\eta_2} (1 + m_2 N) \bmod N^2 \end{aligned}$$

若定义 \otimes 为两个向量对应分量的乘积, 即

$$E(m_1) \otimes E(m_2) = (A_1 A_2, B_1 B_2)$$

而

$$A_1 A_2 = g^{\eta_1 + \eta_2} \bmod N^2, \quad B_1 B_2 = h^{\eta_1 + \eta_2} [1 + (m_1 + m_2)N] \bmod N^2$$

因此

$$E(m_1) \otimes E(m_2) = E(m_1 + m_2)$$

由此可见, Bresson 密码算法具有加同态属性。它与 ElGamal 密码算法同态性质的区别是: 尽管加密 m_1 和 m_2 时选取的随机数完全不同, Bresson 算法仍具有同态性。

数字指纹既可以以加嵌入方式嵌入原始媒体数据中, 又可以以乘嵌入方式嵌入原始媒体数据中, 而乘嵌入可以看成加嵌入的特殊形式。在原始媒体数据的时/空域或变换域, 数字指纹采用加嵌入方式嵌入原始的媒体数据中, 若不考虑感知掩蔽模型, 则嵌入规则为

$$y_i = x_i + w_i, \quad i = 1, \dots, n$$

其中, $X = \{x_1, x_2, \dots, x_n\}$ 为选取的原始载体序列, $Y = \{y_1, y_2, \dots, y_n\}$ 是嵌入指纹后的载体序列, $W = \{w_1, w_2, \dots, w_n\}$ 为嵌入的指纹信号。由 Bresson 公钥密码算法的同态性质可知

$$E(y_i) = E(x_i + w_i) = E(x_i) \otimes E(w_i)$$

文献[33]提出的匿名指纹方案有 4 个参与实体: 销售商(S)、用户(B)、证书机构(CA)、仲裁者(A), 其中 CA 为可信的第三方。该方案包括初始化、指纹嵌入、跟踪与仲裁 3 个子协议。在初始化协议执行阶段, B 根据购买需求, 以真实身份向 CA 提出申请, CA 为 B 生成假名, 并为该次购买行为生成相应的数字指纹。然后, 通过指纹嵌入协议, S 将指纹信息嵌入 B 所购买的媒体数据中, 并将带有指纹的媒体数据发送给用户。尽管由 S 实施指纹的嵌入操作, 由于采用了同态公钥密码算法, S 并不知道嵌入媒体数据中的指纹的具体内容。一旦发现了非法复制的媒体数据, 即可启动跟踪与仲裁子协议, 使得 S 在不需要第三方帮助的情况下能够找到非法分发者。如果被指控的购买用户 B 否认其非法分发行为, 设计的匿名指纹方案使 A 在没有购买用户 B 参与的情况下, 只需 S 提供的证据就可以做出 B 是否无辜的公正仲裁。具体方案请参阅文献[33]。

文献[33]构造的匿名指纹方案避免了常见的匿名指纹方案中, 如安全多方计算或零知识证明等过于复杂的密码协议的使用, 从而使协议的实现变得简单。而且此方案也容易与感知掩蔽模型结合, 从而提高数字指纹的鲁棒性。

参考文献

- [1] N. Wagner. Fingerprinting. Proceedings of the 1983 IEEE Symposium on Security and Privacy, 1983, (8): 18-22.
- [2] D. Boneh, J. Shaw. Collusion-secure fingerprinting for digital data. In: Don Coppersmith. Advances in Cryptology-Proceedings of the Crypto'95. Lecture Notes in Computer Science 963. Springer-Verlag, 1995. 452-465.
- [3] I. Cox, J. Kilian, T. Leighton, T. Shamoon. Secure spread spectrum watermarking for multimedia. IEEE Transactions on Image Processing, 1997, 6(12): 1673-1687.
- [4] E. Zane. Efficient Watermark Detection and Collusion Security. In: Proceedings of the 4th International Conference on Financial Cryptography, Springer-Verlag, 2001. 21-32.
- [5] B. Chor, A. Fiat, M. Naor. Tracing traitors. In Proc. Crypto 1994, Santa Barbara, USA, Aug. 27-31, 1994, LNCS 839, 257-270.
- [6] 刘振华, 尹萍. 信息隐藏技术及应用. 北京: 科学出版社, 2002.
- [7] 吕述望, 王彦, 刘振华. 数字指纹综述. 中国科学院研究生院学报, 2004, 21(3): 289-298.
- [8] J. Kilian, T. Leighton, L. R. Matheson, et al. Resistance of digital watermarks to collusive attacks. Technical Report TR-58598, Department of Computer Science, Princeton. 1998. <http://citeseer.nj.nec.com/kilian98resistance.html>
- [9] F. Ergun, J. Kilian, R. Kumar. A note on the limits of collusion-resistant watermarks. In: Jacques Stern. EUROCRYPT 1999. Lecture Notes in Computer Science 1592. Springer-Verlag, 1999. 140-149.
- [10] F. Zane. Efficient watermark detection and collusion security. Lecture Notes in Computer Science. Springer-Verlag, 2001. 1962: 21-32.
- [11] J. Guth, B. Pfitzmann. Error-and collusion-secure fingerprinting for digital data. Information Hiding. Lecture Notes in Computer Science 1768. Springer-Verlag, 2000. 134-145.
- [12] R. Safavi-Naini, Y. Wang. Collusion secure q-ary fingerprinting for perceptual content. Security and Privacy in Digital Rights Management. Lecture Notes in Computer Science 2320. Springer-Verlag, 2002. 57-75.
- [13] J. Lofvenberg, J. Wiberg. Random codes for digital fingerprinting. Technique Report LiTH-ISY-R-2059 ISSN 1400-3902, Department of Electrical Engineering, Linköping University, 2000.
- [14] W. Trappe, M. Wu, K. J. Ray Liu. Collusion-resistant fingerprinting for multimedia, International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Orlando, FL, USA, 2002.5, 3309-3312.
- [15] T. Lindkvist. Characteristics of some binary codes for fingerprinting. Information Security. In: J Pieprzyk, E Okamoto, J Seberry. Third International Workshop Proceedings, ISW2000. Lecture Notes in Computer Science 1975. Springer-Verlag, 2000. 97-107.

- [16] J.Domingo-Ferrer, J. Herrera-Joancomarti. A simple collusion-secure fingerprinting schemes for images. In: Latifi S. Proceedings of the International Symposium on Information Technology: Coding and Computing (ITCC 2000) . Los Alamitos: IEEE Computer Society Press, 2000. 128-132.
- [17] J. Dittmann, A. Behr, M. Stabenau, et al. Combining Digital Watermarks and Collusion Secure Fingerprints for Digital Images. IS&T/SPIE Conf Security and Watermarking of Multimedia Contents, California, 1999. 171-182.
- [18] J. N. Staddon, D. R. Stinson, R. Wei. Combinatorial properties of frameproof and traceability codes. IEEE Tram. Infom. Theory, 2001, IT-47: 1042-1049.
- [19] 朱岩, 杨永田, 冯登国. 合谋安全的卷积指纹信息码. 软件学报, 2006, 17(7): 1617-1626.
- [20] 王祖喜, 王文宗, 葛强, 胡汉平. 基于残留特征跟踪的抗合谋数字指纹. 软件学报. 2011, 22(8): 1884-1896.
- [21] H. Haken . Synergetic Computers and Cognition-A Top-Down Approach to Neural Nets. Berlin: Springer-Verlag, 1991.
- [22] Z. X. Wang, Q. Ge, W. Z. Wang, H. P. Hu. Digital fingerprinting based on Synergetic. Proceedings of the International Conference on Computational Intelligence and Software Engineering, 2009: 1-4
- [23] B. Pfitzmann, M. Schunter. Asymmetric fingerprinting. In: Ueli M Maurer. Advances in Cryptology-EUROCRYPT' 96. Lecture Notes in Computer Science 1070. Springer-Verlag, 1996. 84-95.
- [24] B. Pfitzmann, M. Waidner. Asymmetric fingerprinting for larger collusions. Int T Matsumoto. Proceedings of the 4th ACM Conference on Computer and Communications Security. New York: ACM Press, 1997. 151-160.
- [25] J.Domingo-Ferrer. Anonymous fingerprinting based on committed oblivious transfer. In: Hideki Imai, Yuliang Zheng. Public Key Cryptography, 99. Lecture Notes in Computer Science 1560. Springer-Verlag, 1999. 43-52.
- [26] N. Memon, P. W. Wong. A buyer-seller watermarking protocol. IEEE Signal Processing Society 1998 Workshop on Multimedia Signal Processing. IEEE, 1998. 291-296.
- [27] P. Tomsich, S. Katzenbeisser. Copyright protection protocols for multimedia distribution based on trusted hardware. In: P Pacyna, Z Papir. Proceedings of Protocols for Multimedia Systems (PROMS 2000). IEEE, 2000. 249-256.
- [28] F. Bao. Multimedia content protection by cryptography and watermarking in tamper-resistant hardware. ACM Multimedia 2000 Electronic Proceedings. <http://www.acm.org/sigmm/mm2000/ep/bao/index.html>
- [29] B. Pfitzmann, M. Waidner. Asymmetric fingerprinting for larger collusions. Proceedings of the 4th ACM Conference on Computer and Communications Security. New York: ACM Press, 1997: 151-160.
- [30] N. Memon N, P. W. Wong . A Buyer-Seller watermarking protocol. IEEE Trans. on Image Processing, 2001, 10(4): 643-649.

- [31] 王彦, 吕述望, 刘振华. 一种基于秘密分享的非对称数字指纹体制. 中国科技大学学报. 2003, 33(2): 237-242.
- [32] 朱岩, 杨永田, 叶志远, 邹维, 冯登国. 购买者透明的非对称扩频指纹方案. 电子学报, 2006, 34(6): 1041-1047.
- [33] 孙中伟, 冯登国, 武传坤. 基于加同态公钥密码体制的匿名数字指纹方案. 软件学报. 2005, 16(10): 1816-1821.
- [34] B. Pfitzmann, M. Waidner. Anonymous fingerprinting. In: Walter Fumy. Advances in Cryptology-EUROCRYPT'97. Lecture Notes in Computer Science 1233. Springer-Verlag, 1997. 88-102.
- [35] B. Pfitzmann, A. R. Sadeghi. Coin-Based Anonymous Fingerprinting. Eurocrypt'99, LNCS1592, Springer-Verlag, Berlin, 1999. 150-164.
- [36] J. Camenisch. Efficient anonymous fingerprinting with group signature. Advances in Cryptology-ASIACRYPT 2000, LNCS 1976, Springer-Verlag 2000, 415-428.
- [37] 李刚. 数字指纹协议的研究与应用. 上海交通大学硕士学位论文, 2005.
- [38] E. Bresson, D. Catalano, D. Pointcheval. A simple public key cryptosystem with a double trapdoor decryption mechanism and its applications. In: Lai CS, ed. Aciacrypt 2003. LNCS 2894, Berlin: Springer-Verlag, 2003. 37-54.

DRM 标准

由于 DRM 自身涉及的层面较多，需求复杂，对于 DRM 系统结构及权限描述语言等关键方面并未达成一致的标准，基本上没有一个统一的、可互操作的技术框架。如前所述，在权利表达语言这个 DRM 关键领域，存在着 XrML 和 ODRL 两大标准之争。而在流媒体、网页、电子出版等 DRM 应用领域，也是百花齐放，存在着多种不同的应用方案与技术路线，目前都没有实现 DRM “互操作性”的要求。

微软的 Windows Media DRM 平台于 1999 年 4 月首次亮相，受到了内容提供商的广泛欢迎，迄今已在超过 5 亿台计算机上安装，应用于 50 多种不同的音乐和视频服务，成为目前用户数最多的 DRM 技术。微软已经将 Windows Media DRM 与 Windows Mobile 操作系统进行捆绑以进军移动通信领域，但由于它是私有标准，因此目前移动通信领域大部分运营商没有计划采用微软的 DRM 标准。

苹果公司的 Apple iTunes DRM 是在其音乐下载平台使用的一种 DRM 私有标准，主要用于苹果公司的 iPod 音乐下载，目前暂无其他应用。

另外，还有很多运营商和厂商为了实现数字版权管理，自行开发了一些私有标准的 DRM 产品，但这些私有标准的 DRM 技术已逐渐退出历史舞台。

OMA 组织在 2002 年 9 月完成并发布了 OMA DRM 1.0 规范，其主要内容是在 OMA 下载业务中，采用 DRM 技术，控制对下载媒体对象的使用。OMA DRM 1.0 规范在安全性方面存在明显的弊端，它假定用户终端本身是可信任的，且权限对象使用明文描述和传输，而且缺乏对内容密钥进行安全管理和传输的基础平台。2006 年 3 月，OMA 发布了 OMA DRM 2.0，制定了基于 PKI 的安全信任模型，给出了移动 DRM 的功能体系结构、权利描述语言标准、DRM 数字内容格式和权限对象获取协议^[1,2,3,4]。OMA 的 DRM 标准主要针对移动终端特性制定，可以认为是目前唯一被标准化的数字版权管理平台。OMA DRM 标准是完全开放的，目前大部分运营商都已经采用 OMA DRM 标准。

我国数字音视频编解码技术标准工作组（AVS 工作组）在数字媒体标准制定方面开展了卓有成效的工作，所制定的《信息技术先进音视频编码》第六部分《数字媒体版权管理》（GB/T 20090.2）于 2006 年 3 月正式实施^[5]。AVS DRM 国家标准的制定是我国在数字媒体版权管理标准方面迈出的重要一步。

5.1 OMA DRM 1.0

OMA DRM 系统定义了媒体对象（Media Object, MO）和权限对象（Rights Object, RO）两个概念，将原始的未经 DRM 系统保护的数据（图片、音乐等）通过成熟的安全加密算法进行加密，加密之后的原始数据称为媒体对象。加密原始数据时使用的密钥和该媒体对象对应的权限信息单独形成独立于媒体对象的媒体对象。权限信息分为两个方面：

- ① 操作形式的限制，包括显示、执行、运行等。
- ② 使用限制信息，包括次数、时间段、累计时间等。

OMA DRM 系统可以使内容提供商为其提供给消费者的媒体对象定义权限对象，每个媒体对象可以对应多个权限对象，相应的每个权限对象具有不同的价格，比如用户可以选择免费预览每个媒体对象，也可以选择为全面使用这个媒体对象支付费用，甚至可以选择支付在一段时间内使用该媒体对象的费用。

OMA 组织认为 DRM 1.0 标准定义的是一个初级的 DRM 系统，将来可以扩展为一个更加复杂和安全的 DRM 系统。

OMA DRM 1.0 定义了三种应用模型，如图 5-1 所示。

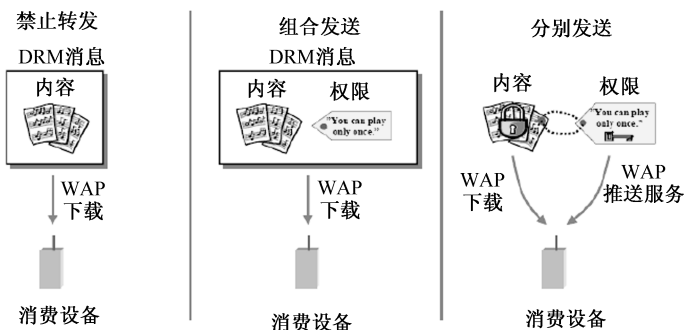


图 5-1 OMA DRM 1.0 的用户实例

1. 禁止转发

一个媒体对象被封装在一个 DRM 消息中传输给设备，允许设备使用内容，但是内容不能转发到其他设备，且设备不能修改此媒体对象。

2. 组合发送

一个权限对象和一个媒体对象被封装在一个 DRM 消息中发送给设备，设备可以根据权限对象的规定向用户提供内容，但不能修改、转发权限对象和媒体对象。

3. 分别发送

内容对象打包成一种特殊的 DRM 内容格式（DRM Content Format, DCF），采用对称加

密技术，必须使用内容加密密钥（Content Encryption Key, CEK）才能够访问媒体内容，CEK 存储在版权对象中。这样，内容可以经过非安全的传输途径传送，而版权对象的传送则需要更安全的传输通道。

4. 超级分发

超级发送是分别分发的一种特殊应用方式。在这种模式下，终端之间可以通过任何可能的途径传送媒体对象，而权限对象只能通过 WAP 推送服务，从权限发布服务器获得。这样就不在任何版权维护的商业模式的条件下，极大地鼓励了用户之间更加方便地共享媒体对象。

OMA DRM 1.0 提出时间较早，虽然在国内外运营商中已经有很多的商用，但是存在较多问题^[6]。

（1）内容安全性问题

在 OMA DRM 1.0 的禁止转发和组合发送模型中，不对数字内容进行加密，数字内容是以原始内容传输的，因此伪终端可以直接从禁止转发和组合发送包中解析出原始数字内容，造成数字内容被盗版。

（2）密钥安全性问题

在 OMA DRM 1.0 的分别发送模型中，数字内容被加密，解密密钥通过相对安全的 WAP Push 方式传送，并且通过权限对象不能离开终端等限制保证密钥的安全性。但因为解密密钥是以明文存放的，万一权限对象离开终端，非法用户就有可能获得解密密钥，从而获得原始内容，造成数字内容被盗版。

（3）时间安全性问题

对于有时间限制的权限对象，系统依赖于终端的时间对权限对象的可用性进行控制，对于某些终端，用户可以修改终端时间，这样容易造成对版权对象的使用失去控制。

（4）缺少计费确认

权限对象是采用非确认机制的推送方式发给用户的。权限对象下发到终端后，需要对用户进行计费，由于没有权限对象是否成功推送的确认机制，因此可能出现终端未成功接收权限对象却扣费的情况。

（5）业务模型单一

OMA DRM 1.0 的目的是针对 OMA 下载业务的，因此对其他业务模型如流媒体、推送内容、MMS（Multimedia Messaging Service，多媒体信息服务）消息的权限管理等并未考虑。

总之，OMA DRM 1.0 是在假定终端安全可信的基础上建立的模型，缺少完备的密钥管理体系，不能对终端进行有效的身份认证和保证设备安全，而且缺少计费确认，业务模型非常单一。因此，OMA DRM 1.0 在 3G 网络中的应用只能是一个过渡方案，3G 业务需要更加成熟的 DRM 解决方案。为了解决 OMA DRM 1.0 的问题，OMA 组织在 2004 年 2 月开始制定 DRM 2.0 规范，2004 年 7 月发布候选版本，2006 年 3 月正式定稿并发布了 DRM 2.0 标准。

OMA DRM 2.0 标准包括 OMA DRM 2.0 规范、体系结构、内容格式、权利描述等^[1,2,3,4]，下面将对 OMA DRM 2.0 标准的主要内容进行介绍。

5.2 OMA DRM 2.0 体系结构

5.2.1 角色定义

在 OMA DRM 系统中，功能实体被定义为不同的角色。功能实体的划分是逻辑上的，并不代表物理网络节点（如服务器）。根据不同的配置，不同的功能实体可能由相同或不同的物理节点来实现，被不同角色所操作。通常，我们在描述 DRM 系统时，采用功能实体而不是角色来描述。

OMA DRM 2.0 包括终端 DRM 代理（DRM Agent, DA）、内容发行者（Content Issuer, CI）、权限对象发行者（Rights Issuer, RI）、用户和移动存储设备等外置存储设备（Off-device Storage）五个功能实体。用户能够通过超级分发等各种方式获得受保护的数字内容，数字内容使用权限通过 ROAP（Rights Object Acquisition Protocol，权限对象获取协议）获取，使用权限与一个或者一组 DA 绑定，数字内容的使用受到了严格的控制。

1. DRM 代理

DA 安装在被 DRM 系统所认证的终端设备上，终端设备的责任是根据权限对象 RO 所规定的权限，控制相应的媒体对象 MO 的操作。使用 DRM 内容的用户仅能通过 DA 访问 DRM 内容，强制执行附带有 DRM 内容上的访问权限控制功能，实现了对 DRM 内容的可控访问。

所有 DRM 代理都有一个唯一的公/私钥对和一个证书。证书包括附加信息，如制造商、设备类型、软件版本、序列号等。证书允许内容发布者和 RI 能安全地认证一个 DRM 代理。

2. 内容发行者

这是分发 DRM 媒体的实体。OMA DRM 规范定义了 DRM 媒体的传输格式，通过这种方式，DRM 媒体内容可以通过不同的传输方式，从内容发行者分发到终端设备。内容发行者可以自己原始的媒体数据进行处理形成 DRM 媒体，也可以从其他地方获得已经打包好的 DRM 媒体。

3. 权限对象发行者

这是分配 DRM 媒体对象权限和限制的实体，并且能够生成权利对象 RO。RO 是使用 XrML 语言表达的文本，包含了其关联的 DRM 媒体的使用权限和限制，DRM 媒体对象在 RO 规定的权限和限制内使用。

DRM 代理和权限对象发行者必须相互认证。ROAP 是 DRM 代理和权限对象发行者之间一系列的安全协议，在终端设备中，DRM 代理发起 ROAP 从权限对象发行者获得 RO，并在

这之前进行双向认证工作。

内容提供商是具有内容发行者和权限对象发行者功能的实体。用户是 DRM 内容的使用者，用户可以通过权限对象发行者访问 DRM 媒体。

5.2.2 OMA DRM 2.0 的基本架构

OMA DRM 2.0 的基本架构如图 5-2 所示^[1]。可以看出，在 OMA DRM 2.0 系统中，内容和权限对象都进行了加密，而且受保护内容可以在用户间随意拷贝。用户得到此内容后，必须获取相关的权限对象才可以使用此内容。

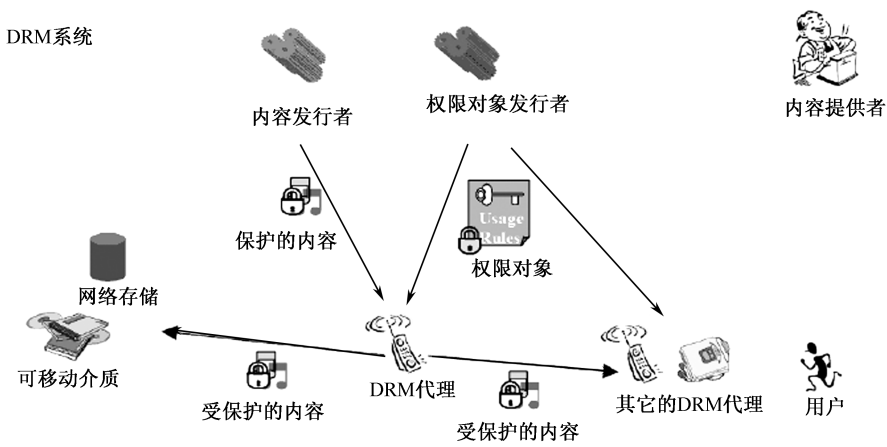


图 5-2 OMA DRM 2.0 的基本架构

OMA DRM 2.0 对 OMA DRM 1.0 进行了大量补充，使版权保护变得更加灵活而有效。它所要求的数字版权管理信任模式基于 PKI（Public Key Infrastructure，公钥基础设施），能对终端的 DRM 代理和权限对象发行者进行双向认证，安全性大大提高；支持的应用场景比较丰富，包括预览、下载 DRM、流媒体 DRM、多媒体消息 DRM、事务跟踪、域管理和与用户标识绑定等；提供了更加灵活、丰富和复杂的商业模式和用户使用模式，如拉（Pull）模式、推送（Push）模式、流模式、超级分发模式、备份和恢复模式、非连接设备支持模式、媒体对象和权限对象的输出模式以及域共享模式。

OMA DRM 2.0 相对于 OMA DRM 1.0 提出了许多新的特性。

1. ROAP

ROAP 是 OMA DRM 2.0 新定义的权限对象发行者 RI 和移动终端 DA 之间的协议，移动终端 DA 和 RI 可以借助 ROAP 更加安全地请求和获取权限对象 RO。该协议的安全交互机制类似于 SSL 协议。

2. 基于 PKI 的安全机制

一种遵循既定标准的密钥管理平台，能够为所有网络应用提供加密和数字签名等密码服务，以及必需的密钥和证书管理体系。

3. 域的概念

OMA DRM 2.0 的域允许权限提供者将权限和内容加密密钥提供给一组 DRM 代理，而不是一个 DRM 代理。这样，属于同一个域的 DRM 代理可以离线共享 DRM 内容，用户可以在其拥有的多个设备上使用 DRM 内容，或者与域内的其他用户共享 DRM 内容。

4. 对流媒体业务的支持

主要是通过定义 PDCF（Packetized DRM Content Format，封装的 DRM 内容格式）来实现对连续传输的流式媒体进行 DRM 的保护机制。

5. 非连接设备的支持

OMA DRM 2.0 允许连接设备作为中介辅助非连接设备来购买和下载内容及权限对象，从而使一些本身没有网络连接功能的终端也能够从移动网络中获得 DRM 内容和权限对象。这一功能的实现基于域共享模式。

6. 丰富的权限功能

OMA DRM 2.0 提供了丰富的权限功能，如组权限功能、复合权限功能、域权限功能、权限继承功能和权限恢复功能。丰富的权限功能为运营商提供了更多的运营策略选择，为用户提供了更加灵活的内容使用方式。

5.2.3 OMA DRM 2.0 工作机制

OMA DRM 2.0 采用独立的功能体系结构，如图 5-3 所示^[7]。

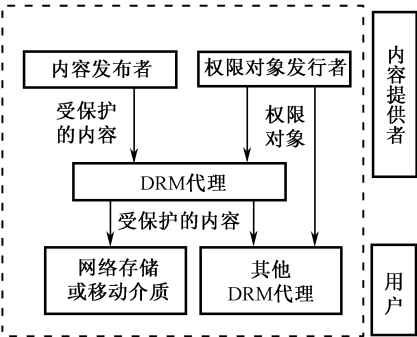


图 5-3 OMA DRM 2.0 功能体系结构

内容发行者 CI 发送 DRM 内容，权限对象发行者 RI 产生一个权限对象 RO 来控制 DRM 内容的使用。OMA DRM 2.0 使 DRM 内容和 RO 逻辑分离，可以分别或同时请求/发送 DRM 内容和 RO。例如，用户选择一个内容并付费，然后在同一个交易中接收 DRM 内容和 RO。若以后 RO 过期，用户无须再次下载 DRM 内容，只需要获取一个新的 RO 即可。

OMA DRM 2.0 实现内容保护的基本步骤如下。

- ① CI 将数字内容进行加密打包后发布，并将内容密钥提供给 RI；
- ② DRM 代理 DA 从 CI 下载获取加密后的内容（解密该内容所需的密钥包含在该用户定购的 RO 中）；
- ③ RI 按用户定购该内容的权限要求生成 RO，并用 DA 公钥对 RO 中的内容密钥进行加密封装；
- ④ DA 通过 ROAP 登记注册到 RI，向 RI 请求 RO；
- ⑤ RI 向 DA 发送 RO，并对 RO 使用 RI 证书私钥进行数字签名以保证 RO 的可靠性和传输完整性；
- ⑥ DA 收到 RO，用 RI 证书公钥验证数字签名以确认 RO 的可靠性和完整性，然后用其证书私钥解密提取内容密钥以及使用该内容的权限；
- ⑦ DA 用提取自 RO 中的内容密钥对内容进行解密，并按相应权限使用解密后的数字内容。

OMA DRM 2.0 详细的工作流程如图 5-4 所示。

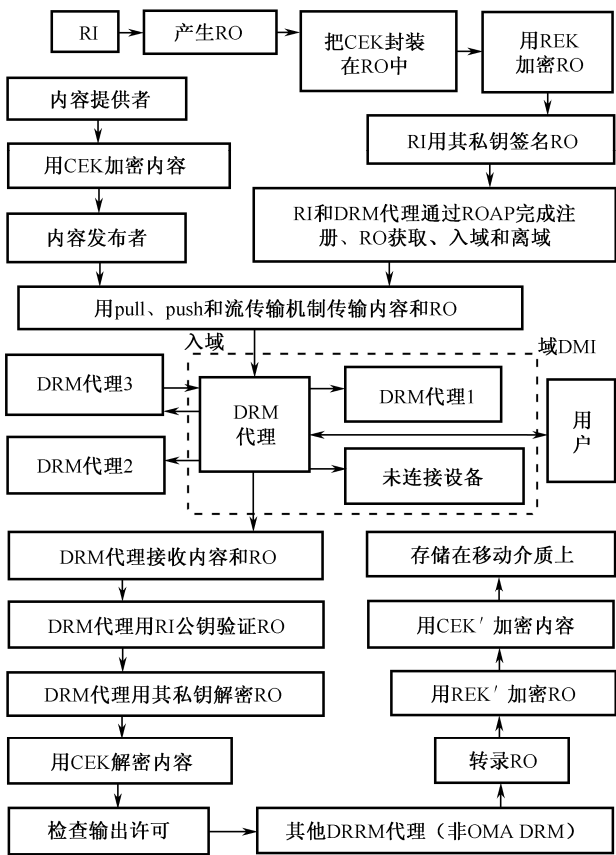


图 5-4 OMA DRM 2.0 工作流程

5.3 ROAP

5.3.1 ROAP 的工作流程

ROAP 是 OMA DRM 2.0 规范的主要内容^[4]，用来完成权限对象发行者（RI）/内容发行者（CI）和设备（DA）之间的注册、权限对象（RO）的获取、入域或离域操作。一个 ROAP 组包含：4 通道注册协议、2 通道 ROAP、1 通道 ROAP、2 通道入域协议和 2 通道离域协议。除 1 通道 ROAP 外，ROAP 组里其他协议都由 ROAP 触发器启动。只有成功执行 4 通道 ROAP 注册协议，建立一个基于设备和 RI 的有效前后关系后，才能进一步执行 RO 获取、入域或离域操作。

4 通道注册协议是一个安全的信息交换和同步交换协议，用于完成权限对象发行者 RI 和设备之间的注册功能。其流程见图 5-5 中的 4 通道注册协议部分。一般只在第 1 层执行此协议，但当 RI 认为设备的 DRM 时间不准确或在交换、更新信息时，也可执行此协议；2 通道 ROAP 是设备用于获取 RO 的一种协议，它包含设备和 RI 间的相互认证、RO 的完整性保护和发送。其流程见图 5-5 中的 2 通道 ROAP 部分。只有成功执行一个 4 通道注册协议，在设备中建立一个基于设备和 RI 的有效前后关系后，才能成功执行此协议；1 通道 ROAP 就是只能从 RI 把 RO 发送给设备（如消息/推），它是 RI 单方面发起的，不需要设备发送任何信息，实际上它是 2 通道变量的最后一个消息。其流程见图 5-5 中的 1 通道 ROAP 部分。只有成功执行一个 4 通道注册协议，在设备中建立一个基于设备和 RI 的有效前后关系后，才能成功执行此协议。2 通道入域协议就是一个设备通过此协议加入一个域，其流程见图 5-5 中的 2 通道入域协议部分。只有成功执行 4 通道注册协议，在设备中建立一个基于设备和管理着域的 RI 的有效前后关系后，才能成功执行此协议。入域协议成功完成后，就会在设备中建立一个域前后关系，它含有详细的域钥信息。设备通过使用一个域前后关系来安装和使用域 RO。2 通道离域协议就是一个设备通过此协议离开一个域，其流程见图 5-5 中的 2 通道离域协议部分。只有成功执行 4 通道注册协议，在设备中建立一个基于设备和管理着域的 RI 的有效前后关系后，才能成功执行此协议。

图 5-5 显示了 ROAP 的工作流程^[7]。

① RI 产生一个用于注册的 ROAP 触发器给 DRM 代理（DA）。

② DA 接到 ROAP 触发器后，会尽可能快地启动 ROAP 协议交换。在启动 4 通道注册协议前，DA 必须获得用户同意，除非 ROAP 触发器的 URL 元素的域名部分和用户同意列表中的登录一致，则不需要获得用户同意就能访问 RI。征得用户同意后，DA 向 RI 发送一个包含设备身份（ID）的设备问候消息来启动 4 通道注册协议。RI 通过<设备问候>元素来识别 4 通道 ROAP 注册协议里的 ROAP-设备问候消息，并通过设备 ID 来识别设备。

③ 为响应 ROAP-设备问候消息，RI 向设备返回一个包含 RI ID 的 4 通道 ROAP 注册协议中的第 2 个消息：ROAP-RI 问候消息。设备上的 DRM 代理必须用<RI ID>元素验证它和 RI 之间有一个有效的 RI 前后关系，并用它识别 RI。

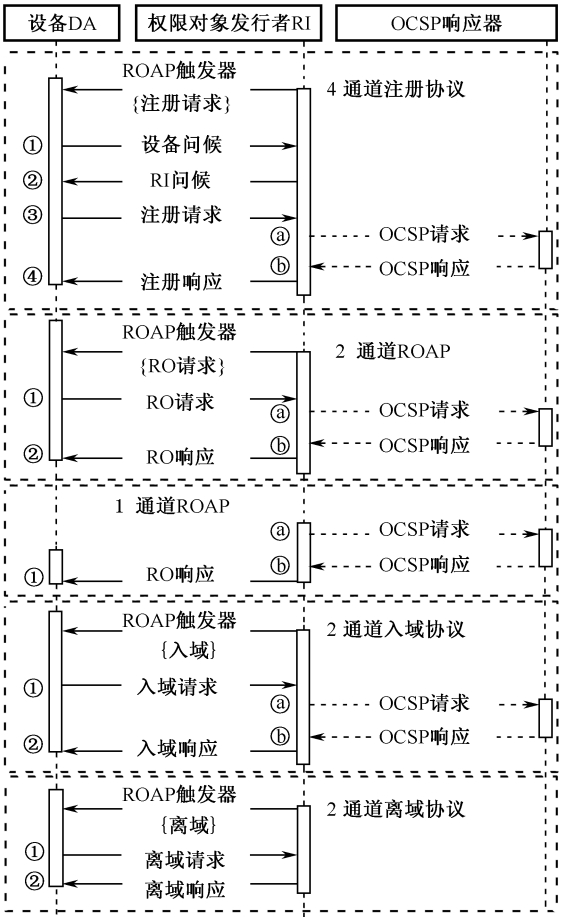


图 5-5 ROAP的工作流程

④ 设备向 RI 发送一个由<注册请求>元素指示的 4 通道 ROAP 注册协议中的第 3 个消息：ROAP-注册请求消息。

⑤ 协议注册期间，RI 为了得到自己的证书，可随时执行一个基于实时的在线证书状态协议（OCSP）请求，再把返回的 OCSP 响应提供给设备；若 RI 认为设备的 DRM 时间不准确，则 RI 要执行一个基于实时的 OCSP 请求；若设备是一个不支持 DRM 时间的未连接设备，则协议注册期间，RI 为了得到自己的证书，必须执行一个基于实时的 OCSP 请求。

⑥ RI 向设备发送一个由<注册响应>元素指示的 4 通道 ROAP 注册协议中的最后消息：ROAP-注册响应消息。若 ROAP-注册请求消息发送成功，则设备再访问 RI 时，可用<ROAP 统一资源定位>元素来发送 ROAP 请求。成功完成协议注册后，就在设备中建立一个 RI 前后关系。

⑦ RI 产生一个用于 RO 请求的 ROAP 触发器给 DA。

⑧ 设备通过<ROAP 统一资源定位>元素向 RI 发送一个由<RO 请求>元素指示的 2 通道 ROAP 中的第 1 个消息：ROAP-RO 请求消息。

⑨ RO 请求期间，RI 为了得到自己的证书，可随时执行一个基于实时的 OCSP 请求，

然后把返回的 OCSP 响应提供给设备；若 RI 认为设备的 DRM 时间不准确，则 RI 要执行一个基于实时的 OCSP 请求；若设备是一个不支持 DRM 时间的未连接设备，则 RO 请求期间，RI 为了得到自己的证书，必须执行一个基于实时的 OCSP 请求。

⑩ RI 向设备发送一个由<RO 响应>元素指示的 2 通道 ROAP 中的第 2 个消息或 1 通道 ROAP 中的唯一一个消息：ROAP-RO 响应消息，它携带着受保护的 RO。

⑪ RI 产生一个用于入域的 ROAP 触发器给 DA。

⑫ 设备向 RI 发送一个由<入域请求>元素指示的 2 通道入域协议中的第一个消息：ROAP-入域请求消息。

⑬ 入域请求期间，RI 为了得到自己的证书，可随时执行一个基于实时的 OCSP 请求，然后把返回的 OCSP 响应提供给设备；若 RI 认为设备的 DRM 时间不准确，则 RI 要执行一个基于实时的 OCSP 请求；若设备是一个不支持 DRM 时间的未连接设备，则入域请求期间，RI 为了得到自己的证书，必须执行一个基于实时的 OCSP 请求。

⑭ RI 向设备发送一个由<入域响应>元素指示的 2 通道入域协议中的第 2 个消息：ROAP-入域响应消息。

⑮ RI 产生一个用于离域的 ROAP 触发器给 DA。

⑯ 设备向 RI 发送一个由<离域请求>元素指示的 2 通道离域协议中的第 1 个消息：ROAP-离域请求消息。

⑰ 为了从域中删除设备，RI 向设备发送一个由<离域响应>元素指示的 2 通道离域协议中的 ROAP-离域响应消息。

5.3.2 域与非连接设备支持

OMA DRM 2.0 允许将 DRM 媒体内容分发到加入域的一组设备，这个域是由权限对象发行者来创建、管理和控制的。一旦域形成且设备已经加入了域，发到域中任意设备的 DRM 媒体对象和权限对象都能被域中的其他设备所共享，而不用再和权限对象发行者联系。设备可以加入任意希望加入的域，接收来自这个域的 DRM 媒体内容^[1]。

1. 域管理

图 5-6 显示了 OMA DRM 2.0 的域管理^[1]。

① 设备 D1、D2、D3 分别与权限对象发行者联系完成加入域 DM1 的注册过程。

② 设备 D1 与权限对象发行者联系，获得 DRM 媒体内容 DCF1 和相对应的域权限对象 RO1。由于设备 D1 是域 DM1 的一个设备。因此，在 D1 上该 DRM 媒体内容和权限对象有效。

③ 设备 D1 将 DCF1 和 RO1 转发给域中的其他设备 D2、D3。因为 D2、D3 是域 DM1 中的设备，因此，这两个设备可以立即使用这个 DRM 媒体对象而不再与权限对象发行者联系。

④ DCF1 同样被转发给设备 D4。但设备 D4 没有加入域 DM1，因此 D4 不能使用 DCF1。

⑤ D4 的用户可以选择与权限对象发行者联系，申请加入域 DM1 并获得访问 DCF1 的权限。由于域是由权限对象发行者管理的，那么权限对象发行者可以决定域的组成成员并决定

D4 是否加入域 DM1。

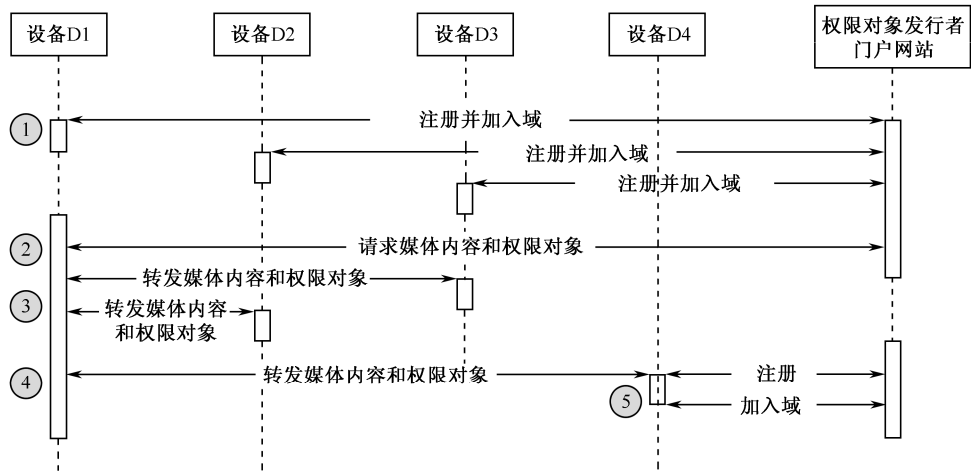


图 5-6 OMA DRM 2.0 的域管理

2. 对非连接设备的支持

通过域功能，OMA DRM 2.0 可以支持非连接设备。DRM 媒体内容和相应的权利对象可以通过连接设备分发给非连接设备。在这种模式中，连接设备和非连接设备必须属于同一个域。

图 5-7 显示了 OMA DRM 2.0 对非连接设备的支持^[1]。

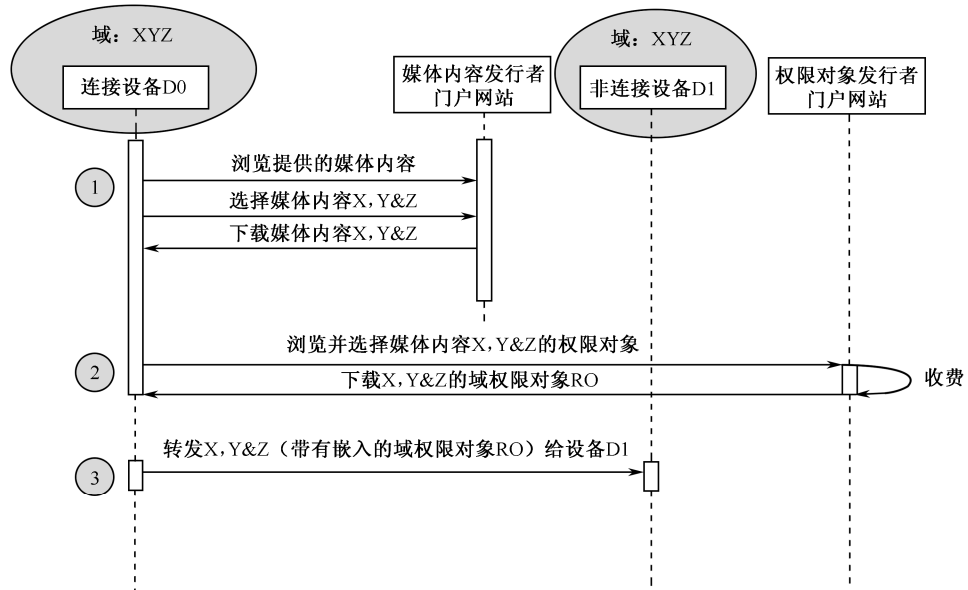


图 5-7 OMA DRM 2.0 对非连接设备的支持

第①步和第②步组成了基本下载的整个过程（pull 模式），连接设备访问内容发行者的门户网站，在选择 DRM 媒体内容之后，DRM 媒体内容 X、Y、Z 被下载到连接设备上，然后，连接设备将与权限对象发行者进行联系，获得 DRM 媒体内容 X、Y、Z 对应的域权利对象，连接设备将这些域权限对象嵌入相对应的 DCF 中。第③步，连接设备转发 DRM 媒

体内容 X、Y、Z（这些 DRM 媒体内容都拥有了嵌入的权利对象）到非连接设备，通过本地的通信连接。

5.3.3 超级分发

DRM 客户端可以通过不同方式转发 DRM 媒体对象到其他设备，如移动存储设备。这些 DRM 媒体对象都是经过加密的对象，接收到 DRM 媒体对象的设备在获得相对应的权限对象之前，是不能使用这些 DRM 媒体对象的。接收设备会从接收到的 DRM 媒体对象的头定义中获得权利对象发行者的 URL，然后与权限对象发行者取得联系并获得相对应的权利对象。图 5-8 显示了超级分发的整体流程^[1]。

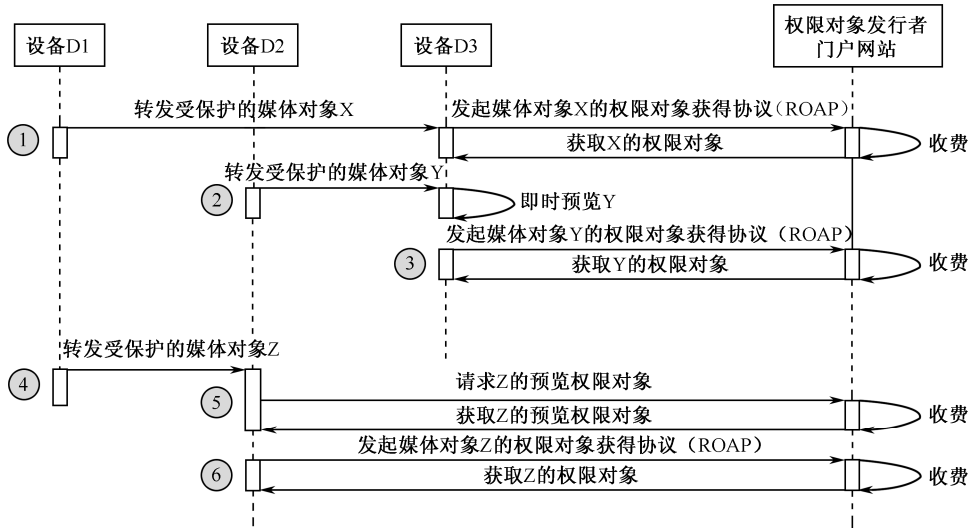


图 5-8 OMA DRM 2.0 的超级分发

第①步包含了超级分发的全部过程。设备 D1 已经获得了 DRM 媒体对象 X 并在本地进行了存储。D3 设备通过本地连接或者移动存储设备与 D1 设备共享 DRM 媒体对象，通过该 DRM 媒体对象头获得权限对象发行者的 URL，发起权利对象获得协议（ROAP）与权利对象发行者取得联系。在完成这个协议并进行付费之后，设备 D3 获得了 DRM 媒体对象 X 相对应的权利对象。

第②和第③步组成了超级分发过程的另外一个用户实例，设备 D2 将 DRM 媒体对象 Y 转发给设备 D3。DRM 媒体对象拥有预览（Perview）头并能为该 DRM 媒体对象提供即时预览权限。设备 D3 能预览使用该 DRM 媒体对象 Y，D3 设备的用户也可以决定是否购买该 DRM 媒体对象 Y 相应的权限对象。当其决定购买权限对象时，设备 D3 将发起权利对象获得协议（ROAP）与权限对象发行者取得联系。在完成这个协议并进行付费之后，设备 D3 获得了 DRM 媒体对象 Y 相对应的权限对象。

第④步和第⑤步组成了与第二步和第三步相类似的用户实例。不同之处在于 DRM 媒体对象 Z 不具有预览（preview）权限，因此，设备 D2 必须与权限对象发行者取得联系并获得相对应的权限对象，才能使用该 DRM 媒体对象。

5.3.4 流媒体的支持

为了分发被保护的流媒体数据，设备必须从内容提供者处获得该流媒体的令牌（token）并且在权限对象的限制下访问流媒体。客户端在获得流媒体头之后，必须与权限对象发行者取得联系，获得相应的权限对象。该权限对象为客户端提供了必要的信息去解码流媒体并播放流媒体。流媒体令牌是一些信息，媒体播放器可以使用这些信息获得流媒体的地址和其他一些参数，并能根据这些信息建立和开始流媒体的分发过程（图 5-9）^[1]。

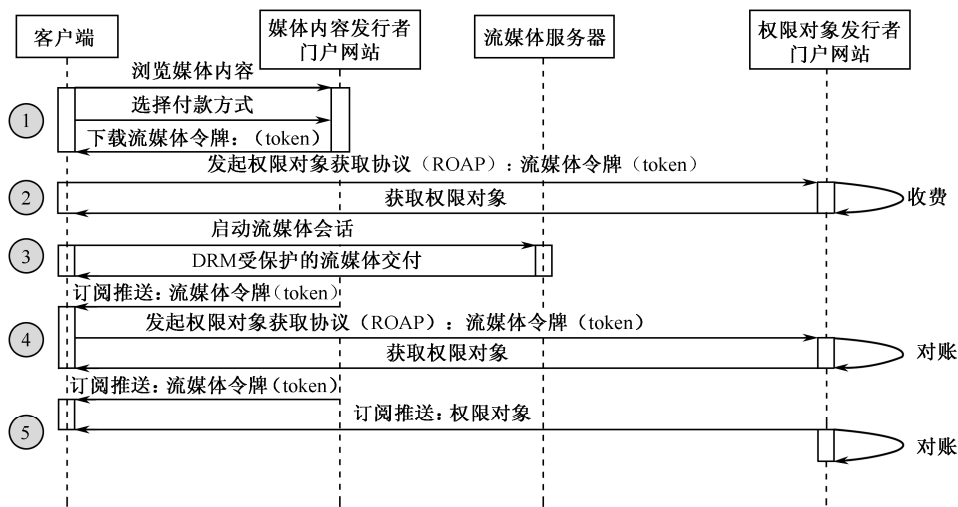


图 5-9 OMA DRM 2.0 的流媒体支持

5.4 OMA DRM 2.0 内容格式

在 OMA DRM 系统中，原始数据被加密打包为特定格式，这种格式称为 DRM 内容格式。在 OMA DRM 2.0 规范中，定义了两种内容格式：DCF（DRM Content Format，DRM 内容格式），用于表示离散型媒体，比如静态图片；PDCF（Packetized DRM Content Format，封装的 DRM 内容格式），用于表示连续型媒体，比如音频、视频。这两种 DRM 内容格式都使用同一种数据结构表示，这种数据结构是基于 ISO Base Media File Format 表示的（ISO 14496-12）。在 DRM 内容格式中，需要表达被加密的媒体对象、原始媒体对象的类型、该媒体对象的唯一标识（方便与权限对象关联）、权限对象发行者 RI 的信息、一些扩展信息^[3]。

5.4.1 基础数据结构定义

ISO Base Media File Format 是基于面向对象思想设计的数据箱（Box）。最基础的 Box 包含两个强制参数：size 和 type。type 标识与该 Box 关联的特定的数据定义，该类型标识是唯一标识数，为 4 字节，称为 4CC（4 Character Code）。size 标识该 Box 的长度。这种格式采用

SDL（Syntax Description Language，语法描述语言）表示，该语言与大多数编程语言极为相似且支持面向对象。OMA DRM 2.0 的内容对象数据定义如表 5-1 所示。

表 5-1 OMA DRM 2.0 的内容对象数据定义

DCF		PDCF	
4CC	用途	4CC	用途
'ohdr'	Common Headers Box	'grpi'	Group ID Box
'mdri'	Mutable DRM Information Box	'mdri'	Mutable DRM Information Box
'grpi'	Group ID Box	'odtt'	Transaction Tracking Box
'odtt'	Transaction Tracking Box	'odrb'	Rights Object Box
'odrb'	Rights Object Box	'ohdr'	Common Headers Box
'odcf'	File Brand	'adaf'	Access Unit Format Box
'odrm'	OMA DRM Container Box	'odkm'	OMA DRM scheme type, OMA DRM scheme information box identifier
'odhe'	Headers box for the Discrete Media profile Box		
'icnu'	Icon URL		
'info'	Info URL		
'odda'	Content Object Box		

Box 类是所有数据结构类的超类，它的数据结构定义如下^[3]：

```
aligned(8) class Box(unsigned int(32) boxtype, optional unsigned int(8) extended_type) {
    unsigned int(32) size;
    unsigned int(32) type=boxtype;
    if(size==1){
        unsigned int(64) largesize;
    }
    else if(size==0){
        //box extends to end of file
    }
    if(boxtype=="uuid"){
        unsigned int(8)[16] usertype=extended_type;
    }
}
```

由于 DCF 具有可扩展性，因此，每一个数据类型都带有版本信息是非常重要的。ISO 规范预先定义了 Fullbox 结构以支持这一需求，这个类来源于 Box 类型^[3]。

```
aligned(8) class FullBox(unsigned int(32) type, unsigned int(8) v, bit(24) f) extends Box(type){
    unsigned int(8) version=v;
    bit(24) flags=f;
}
```

Common Boxes 定义了 DRM 数据各种公用的数据结构。

1. Common Headers Box

Common Headers Box 定义了 DCF 和 PDCF 的头结构。终端设备不能修改该结构中的任何数据^[3]。

```
aligned(8) class OMADRMCommonHeaders extends FullBox('ohdr', version, 0) {
    unsigned int(8) EncryptionMethod;           // 加密方法
    unsigned int(8) PaddingScheme;               // 填充类型
    unsigned int(64) PlaintextLength;            //明文内容长度
    unsigned int(16) ContentIDLength;            // ContentID 字段长度
    unsigned int(16) RightsIssuerURLLength;      // 权限对象发行者 RI 的 URL 字段长度
    unsigned int(16) TextualHeadersLength;       // TextualHeaders 数组的字段长度
    char ContentID[];                           // 内容 ID
    char RightsIssuerURL[];                     // RI 的 URL
    string TextualHeaders[];                    // 附加头部的名值对
    Box ExtendedHeaders[];                      // 扩展头
}
```

2. Textual Headers

TextualHeaders 字段可以包含数据内容的附加信息。Textual Headers 结构使用名值对表示，名字和数值使用 “:” 分割并以 NULL (“\0”) 作为结束标志，下面是 Textual Headers 的一个例子^[3]。

```
Silent:on-demand;http://myissuer.com/silent?cid=428\0Preview:instant;cid:429@myissuer.com\0
```

Textual Headers 头结构主要包括以下几类（如表 5-2）。

表 5-2 OMA DRM 2.0 Textual Headers 结构头定义

名 称	定 义	注 释
Silent	Silent = "Silent" ":" silent-method ";" parameter silent-method = token parameter = silent-rights-url silent-rights-url = token	表示客户端可从 RI 处获得 DRM 内容，不需要用户交互付款
Preview	Preview = "Preview" ":" preview-method (";" parameter) preview-method = token parameter = preview-element-uri preview-rights-url preview-element-uri = token preview-rights-url = token	表示客户端可以预览 DRM 内容
ContentURL	ContentURL = "ContentURL" ":" content-url content-url = token	提供了获取 DCF 或 PDCF 的地址
Content Version	ContentVersion="ContentVersion" ":" original-content-identifier ":" version-identifier original-content-identifier = token version-identifier = *digit	定义了 DRM 内容对象遵从的规范版本

续表

名 称	定 义	注 释
Content-Location	ContentLocation = "Content-Location" ":" content-uri content-uri = token	提供一个内容对象的相关地址，可以用来引用 DCF 文件，或输出内容对象时定义一个有意义的文件名
Custom	OtherHeader = Header-name ":" Header-value Header-name = token Header-value = token	Custom 采用 UTF-8 编码

3. Extended Header

ExtendedHeaders 字段可以包含 0 个或多个嵌套的数据结构，为 Common Header 增加功能。

ExtendedHeaders 字段可以包含一个 OMADRMGroupID 实例^[3]。

```
aligned(8)class OMADRMGroupID extends FullBox('grpi',version,0){
    unsigned int(16)GroupIDLength; //Group ID URI 的长度
    unsigned int(8)GKEncryptionMethod; //Group Key 加密算法
    unsigned int(16)GKLength; //密钥长度
    char GroupID[GroupIDLength]; //Group ID URI
    byte GroupKey[GKLength]; //密钥和加密信息
}
```

4. Mutable DRM Information Box

这个结构中提供了一些信息可供客户端设备编辑。基本定义如下^[3]：

```
aligned(8)MutableDRMInformation extends Box('mdri') {
    Box data[]; // box 数组或自由空间数组
}
```

5.4.2 DCF

这一节定义离散型媒体的 DRM 内容格式 DCF。DCF 的结构如图 5-10 所示^[3]，这个结构必须包含文件头和 OMA DRM 容器。

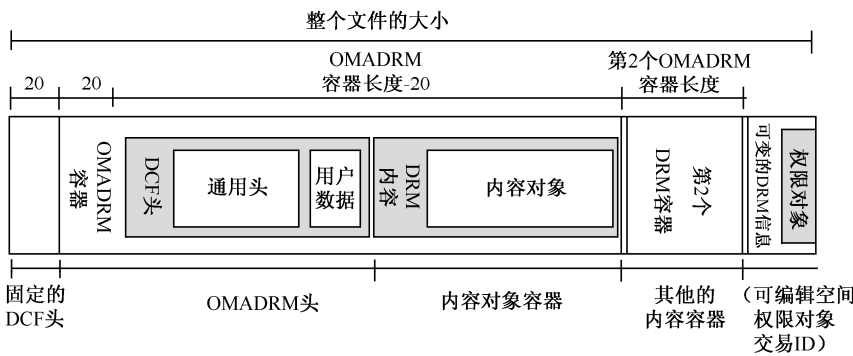


图 5-10 DCF 的结构

DCF 文件的开始是 20bit 固定长度的 DCF 文件头；接着是第 1 个 OMA DRM 容器，必须包含一个 DCF 头和一个受保护的 DRM 内容；DCF 头是偏移 OMA DRM 容器开头 20bit 的第 1 个 Box，包含通用头和用户数据；在 DRM 内容后面，是 OMA DRM 定义的扩充部分，是其他 OMA DRM 容器和可变 DRM 信息；可变 DRM 信息又包含交易 ID 和权限对象。

1. DCF 文件头

DCF 文件头含有商标号和版本域的文件类型 Box。

2. OMA DRM 容器

OMA DRM 容器 OMADRMContainer 必须出现在最顶层，而不能嵌入到别的数据类型中。DCF 文件至少要包含一个 OMADRMContainer，其定义如下：

```
aligned(8) class OMADRMContainer extends FullBox('odrm', version, 0) {
    OMADRMDiscreteHeaders ContentHeaders; // DCF 头
    OMADRMContentObject DRMContent; // 实际的加密内容
    Box Extensions[]; // 扩展部分，在最后的 Box 中
```

OMADRMContainer 必须包含一个 DCF 头 OMADRMDiscreteHeaders 和一个 DRM 内容 OMADRMContent，后面是可选的扩展项。

```
aligned(8) class OMADRMDiscreteHeaders extends FullBox('odhe', version, flags) {
    unsigned int(8) ContentTypeLength; // 内容类型 ContentType 字段的长度
    char ContentType[]; // 内容类型字符串，指定内容对象的 MIME 媒体类型
    OMADRMCommonHeaders CommonHeaders; // 普通头 (和 PDCF 一样)
    if(flags & 0x000001) {
        UserDataBox UserData; // ISO 用户数据 Box (可选)
    }
}
```

3. 通用头

通用头必须出现在 DCF 头中，其定义 OMADRMCommonHeaders 已在 5.4.1 中介绍。

4. 用户数据

当 DCF 包含 UserDataBox 用户数据时，它必须紧跟在 OMADRMCommonHeaders 普通头后面，并用标志 0x000001 指明。

5.4.3 PDCF

PDCF 格式针对类似于音频和视频的媒体内容。音乐和视频文件可以采用 DCF 来封装，但由于 PDCF 格式是特别为连续型媒体设计的，因此它能提供给这类媒体类型更多的优势。PDCF 可以用于下载内容或托管流媒体内容，OMA DRM 为流媒体服务指定了文件格式和附加信息的通用数据结构。在 OMA DRM 2.0 规范中定义了密钥管理功能以支持连续型媒体，

但流媒体服务可以在它们自己的结构中对协议进行优化和编码。对 PDCF 格式的支持在客户端设备是可选的。

图 5-11 显示了 PDCF 的结构^[3]，定义了 PDCF 的关键部分。

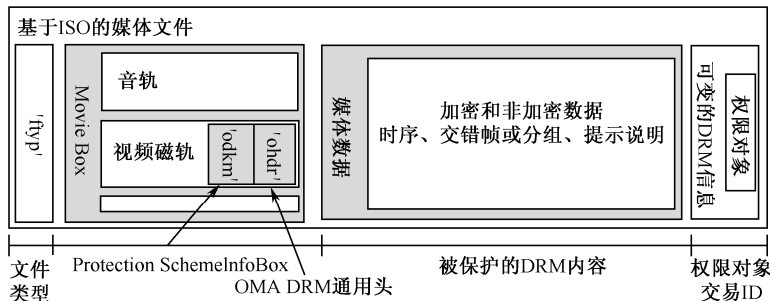


图 5-11 PDCF 的结构

图 5-11 显示了被保护的信息在 PDCF 中的存储。流式 PDCF 和非流式 PDCF 有一点区别。流式 PDCF 必须符合文件格式规范，并且媒体数据以数据包的形式存储。而非流式 PDCF 的媒体数据以特定的格式存储（Sample），访问单元由一个或多个 Sample 组成。

文件格式除了支持 OMA DRM 2.0 外，还支持其他的密码管理系统，在加密的访问单元格式中要指定 OMA DRM。加密过程中，非流式 PDCF 必须在每个访问单元前插入 OMADRMHeader。

1. DRM 方案类型

SchemeTypeBox 包含了 DRM 系统管理密钥和解密内容的信息。由于媒体文件格式也可以支持除 OMA DRM 外的其他密钥管理系统，因此所使用的密钥管理系统在 SchemeType 字段由一个 4CC 指定[ISO14496-12]。

对于符合规范的 PDCF 文件，SchemeType 必须是由 4CC 表示的 'odkm'，并且 SchemeVersion 必须是 0x00000200（2.0 版本）。如果指定 OMA DRM 密钥管理方案为 'odkm'，那么文件为 PDCF，并且必须至少包含一个 OMADRMKMSBox。一个 PDCF 必须仅支持 OMA DRM 的密钥管理系统。

2. 方案信息

SchemeInformationBox ('schi')用于携带 DRM 密钥管理系统的具体信息，因此它只是一个容器 Box。对 DRM OMA，这个 Box 必须包含且仅包含一个 OMADRMKMSBox，作为它的子 Box。

3. OMA DRM 密钥管理系统

在一个 PDCF 文件中，可能有几个 OMADRMKMSBox 实例。

```
aligned(8) class OMADRMKMSBox extends FullBox('odkm', version, 0) {
    OMADRMCommonHeaders Headers; // 通用头，见 5.4.1 节定义
    OMADRMHeaderFormat AUFormat; // 可选
}
```

OMADRMKMSBox 必须包含一个 OMADRMCommonHeaders 通用头，其定义见 5.4.1 节。还可以包含一个 OMADRMFormatBox 作为第 2 个子 Box，用于指定媒体访问单元中头的格式。

```
aligned(8) class OMADRMFormatBox extends FullBox('odaf', 0, 0) {
    bit(1) SelectiveEncryption;
    bit(7) reserved;
    unsigned int(8) KeyIndicatorLength;
    unsigned int(8) IVLength;
}
```

4. 访问单元格式

访问单元格式指定了被 OMA DRM 保护的每个访问单元的格式。一个媒体文件格式指定了媒体数据的特定格式（Sample），但是加密/解密过程要求在每个单元携带附加信息，附加信息依赖于所用的 DRM 密钥管理。OMA DRM 指定它自己的访问单元头，在每个访问单元中它必须放在 Sample 的媒体数据前。

```
aligned(8) class OMADRMHeader {
    bit(1) EncryptedAU; // 访问单元的加密标志。0：访问单元没有加密；1：访问单元被加密
    bit(7) reserved; // 必须为 0
    if (EncryptedAU==1) {
        //在这个版本中，KeyIndicator 的长度为 0，因此这个字段可以忽略
        unsigned int(8 * KeyIndicatorLength) KeyIndicator;
        unsigned int(8 * IVLength) IV; //IV 数据
    }
}
```

当加密 PDCF 内容时，OMADRMHeader 信息必须附加在被处理的访问单元中，同时 OMADRMCommonHeaders 中 EncryptionMethod 字段设置为 NULL。一个播放设备采用这种头信息来进行解密并提取实际的数据。

流媒体格式的 DRM 内容是 PDCF 和标准流媒体格式的融合体，被加密的内容通过实时流媒体协议传输，传输内容中包含了原始载荷。这种被加密的载荷包可以使用任何的流媒体服务传输，比如 RTSP 流媒体协议（RFC2326）、SDP 流媒体协议（RFC2327）和 RTP 传输协议（RFC3550）。

5.5 OMA DRM 2.0 权利描述

OMA DRM 2.0 权利描述语言在 ODRL（Open Digital Rights Language，开放数字权利语言）的基础上，对权限对象进行了语法和语义的定义。而 ODRL 本身缺少语义的定义，所以 OMA DRM 2.0 权利描述语言又对 ODRL 的使用提供了补充说明。OMA DRM 2.0 权利描述语言以 XML 方式定义了对 OMA DRM 内容的各种访问许可（如播放、显示、复制、保存）和限制（如次数、时段、质量）^[7]。

OMA DRM 2.0 的权限描述语言根据不同的功能域定义了不同的权利模型，这些模型中包

含了不同的许可和约束。权利模型分别命名为基础模型（Foundation Model）、协议模型（Agreement Model）、上下文模型（Context Model）、许可模型（Permission Model）、约束模型（Constraint Model）、继承模型（Inheritance Model）和安全模型（Security Model）。每个模型都包含由服务商为媒体内容定义的一些许可和约束值。权利对象分层次地将这些属于不同模型的元素合并起来，能够保证内容被安全地操作。每个用户的操作必须遵守这些许可和约束的描述，否则这些操作会被认为是非法操作，将会被拒绝。下面给出主要模型的详细定义^[2]。

1. 基础模型（Foundation Model）

基础模型构成了权限的基础。它包含<right>元素，汇集元信息和协议的信息。基础模型作为起点，包含协议模型和上下文模型。

```
<!ELEMENT o-ex:rights (o-ex:context, o-ex:agreement)>
```

在规范中，<right>元素是所有权限对象的根元素，它强制包含<context>元素和 <agreement>元素，把资产链接到相应的权限。

```
<!ATTLIST o-ex:rights o-ex:id ID #REQUIRED>
```

<right>元素的o-ex:id属性在一个被保护的权限对象环境中识别<right>元素，o-ex:id属性的值可以在<roap:ROPayload>元素中，通过<signature>的<ds:Reference>元素使用。

2. 协议模型（Agreement Model）

协议模型表示对DRM内容所授予的权限，它包含与一组权限相联系的<agreement>元素，对相应的DRM内容指定<asset>元素。协议模型包含许可模型和安全模型。

```
<!ELEMENT o-ex:agreement (o-ex:asset+, o-ex:permission*)>
```

<agreement>元素指定对相应的 DRM 内容所授予的权限，它包含一个或多个<asset>元素和 0 个或多个<permission>元素。

```
<!ELEMENT o-ex:asset (o-ex:context?, o-ex:inherit?, o-ex:digest?, ds:KeyInfo?)>
```

资产<asset>元素指定了DRM内容的身份，该内容由包含在<agreement>元素中的<context>子元素来管理。可选的<inherit>元素使DRM代理由继承的权限对象来确定<inherit>元素中资产的使用权限。注意，在继承的情形中，如果权利对象是作为父权利对象，<KeyInfo>元素应该省缺。可选的<digest>元素对引用的DRM内容提供完整性保护。如果授权，可选的<KeyInfo>元素对DRM内容提供功能访问。<asset>元素使表达式通过其“id”和“idref”属性链接，使得在同一权限对象的其他资产中可以重复使用对一个资产定义的许可证（permission）。如果<asset>元素包含在<permission>元素中，它必须包含一个“idref”属性，而且必须是空的，也就是说，所有的它的可选子元素必须省缺。

```
<!ATTLIST asset o-ex:id ID #IMPLIED>
```

“id”属性是<asset>元素的标识符，它唯一确定每个<asset>。如果<asset>元素是<agreement>元素的子元素，那么只能使用“id”属性；如果<asset>元素是<permission>元素的子元素，则不允许使用；如果没有从同一权限对象的其他地方引用<asset>元素，则应省缺。在权限对象中“id”属性必须是唯一的，并且不允许和<uid>元素中的内容ID相同。

```
<!ATTLIST asset o-ex:idref IDREF #IMPLIED>
```

“idref”属性引用<asset>元素的标识符。如果<asset>元素是<permission>元素的子元素，

那么只能使用“idref”属性；如果<asset>元素是<agreement>元素的子元素，则不允许使用；如果它不引用在同一权利对象中的另一个<asset>元素，则应省缺。

3. 上下文模型（Context Model）

上下文模型提供了权限的元信息。它通过对附加信息的表示扩展了基础模型、协议模型和约束模型。<context>元素在<rights>元素、<asset>元素、<individual>元素、<system>元素和<inherit>元素中使用。由于模型的名称已给定，它的子元素的语义依赖于它所在的权利对象的上下文。

```
<!ELEMENT o-ex:context (o-dd:version?, o-dd:uid*)>
```

<context>元素包含可选的<version>和<uid>元素，它提供了在其父元素中使用的上下文敏感信息，其语义依赖于父元素。<context>元素不允许包含多于一个<uid>元素，除非<context>元素包含在<individual>元素中。

```
<!ELEMENT o-dd:version (#PCDATA)>
```

如果<version>元素的父元素<context>包含在<rights>元素或<system>元素中，则<version>元素应该只出现一次。如果其父元素<context>包含在<rights>元素元素中，那么<version>元素指定权限对象的版本，在这个规范中，其值必须是“2.0”（没有括号）。如果其父元素<context>包含在<system>元素中，那么指定其他DRM系统或内容保护方案的版本，DRM内容和权限对象将输出到那里。

```
<!ELEMENT o-dd:uid (#PCDATA)>
```

如果其父元素<context>包含在<rights>元素中，<uid>元素构成权限对象的标识符，其值必须匹配ROPayload 类型<ro>元素的“id”属性（详见第5.3.9节）。如果其父元素<context>包含在<asset>元素中，<uid>元素指定了对应的DRM内容的标识符，它包含DCF的ContentID值。

4. 许可模型（Permission Model）

许可模型由协议模型扩展而来，通过指定对设备的访问授权，它更加方便地表达了资产的许可信息。许可模型和约束模型相结合，可以提供非常细致的媒体内容的操作控制，许可集包括<play>、<display>、<execute>、<print>和<export>等，使用 DRM 内容必须严格遵守权限对象中许可信息规定的内容。

```
<!ELEMENT o-ex:permission (o-ex:constraint?, o-ex:asset*, o-dd:play?, odd:display?,  
o-dd:execute?, o-dd:print?, oma-dd:export?)>
```

<permission>元素包含一个可选的<constraint>元素，0个或多个<asset>元素和一个可选的许可集，指定一个内容块的权限，如 <play>、<display>、<execute>、<print>和<export>许可元素。

```
<!ELEMENT o-dd:play (o-ex:constraint?)>
```

<play> 元素授予许可来创建一个音频或视频内容的瞬态表示，它包含一个可选的<constraint>元素，顶层<constraint>元素和其子元素指定授予播放权限的DRM代理。

```
<!ELEMENT o-dd:display (o-ex:constraint?)>
```

<display>元素授予许可来制作一个内容瞬态可见的呈现，它包含一个可选的<constraint>元素，如果指定了<display>元素，DRM代理必须由顶层<constraint>元素和其子元素授予显示

权限。

```
<!ELEMENT o-dd:execute (o-ex:constraint?)>
```

`<execute>`元素授予许可来执行简单的可计算元素，它包含一个可选的`<constraint>`元素，如果指定了`<execute>`元素，DRM代理必须由顶层`<constraint>`元素和其子元素授予执行权限。

```
<!ELEMENT o-dd:print (o-ex:constraint?)>
```

`<print>`元素授予许可来创建一个固定的（即静态的）直接可感知的内容表示，它包含一个可选的`<constraint>`元素，如果指定了`<print>`元素，DRM代理必须由顶层`<constraint>`元素和其子元素授予打印权限。

```
<!ELEMENT oma-dd:export (o-ex:constraint)>
```

`<export>`元素授予对DRM内容和相应的权限对象的输出权限，它包含一个强制的`<constraint>`元素，DRM代理必须由顶层`<constraint>`元素和其子元素授予输出权限。

```
<!ATTLIST oma-dd:export oma-dd:mode (move | copy) #REQUIRED>
```

如果`mode`属性等于“move”，`<export>`元素中的`<constraint>`元素可以有`<datetime>`元素，不允许有`<interval>`元素、`<count>`元素、`<accumulated>`元素和`<individual>`元素。

如果输出权限对象而且`mode`属性值等于“move”，DRM代理必须输出包含`<export>`元素的原始权限对象，如果是一个有状态的权限对象，则携带状态信息；在执行输出后，必须使包含`<export>`许可的原始权限对象永久不可用在原始设备上。

如果`mode`属性等于“copy”，`<export>`元素中的`<constraint>`元素可以有`<count>`元素、`<datetime>`元素、`<interval>`元素，不允许有`<accumulated>`元素和`<individual>`元素。

如果`mode`属性等于“copy”，DRM代理必须输出包含`<export>`元素的原始权限对象，如果是一个有状态的权限对象，则不带状态信息；在执行输出后，必须在原始设备上留下未改变的包含`<export>`许可的原始权限对象。

5. 约束模型（Constraint Model）

通过提供了细致的内容消费控制，约束模型扩展了许可模型。约束每次关联到一个许可元素，对于一个授权许可，它所有的约束条件必须满足。如果约束条件不能被终端设备理解或执行，那么这个约束所关联的许可元素将无效或不允许被授权。如果存在`<constraint>`元素，则`<constraint>`元素应包含至少一个子元素。如果`<constraint>`元素没有包含任何约束，如`<count>`、`<datetime>`等，那么这个元素就不受约束，根据包含这样不受约束`<constraint>`元素的许可，DRM代理可以不受约束地访问DRM媒体内容。

```
<!ELEMENT o-ex:constraint (o-dd:count?, oma-dd:timed-count?, o-dd:datetime?, odd:interval?,  
o-dd:accumulated?, o-dd:individual?, oma-dd:system*)>
```

`<constraint>`元素是约束模型的最顶层元素，它包含可选的`<count>`、`<timed-count>`、`<datetime>`、`<interval>`、`<accumulated>`、`<individual>`和`<system>`元素。只有当其父`<permission>`元素包含`<export>`、`<play>`或`<display>`元素时，`<constraint>`元素才包含`<system>`元素。

```
<!ELEMENT o-dd:count (#PCDATA)>
```

<count>元素指定许可对资产授权的次数，它包含一个正整数值。如果其父<constraint>元素包含在<export>元素中，<count>元素指定<export>元素对DRM内容和权限对象本身授权许可的次数。

<!ELEMENT oma-dd:timed-count (#PCDATA)>

<timed-count>元素的语义和<count>元素一样，但它附加了一个可选的timer属性。

<!ATTLIST oma-dd:timed-count oma-dd:timer CDATA #IMPLIED>

timer属性包含一个正整数值，在它指定的秒数后，由<timed-count>元素的值指定的计数状态开始递减。例如，timer的值设为30，timed-count设为5，相应的媒体对象可以呈现5次。在内容被呈现30秒后，剩余的访问数量递减。

<!ELEMENT o-dd:datetime (o-dd:start?, o-dd:end?)>

<datetime>元素指定时间范围，它包含可选的<start>和<end>元素。

<!ELEMENT o-dd:start (#PCDATA)>

<start>元素指定开始时间/日期，其值的一般形式在[ISO8601]中定义。

<!ELEMENT o-dd:end (#PCDATA)>

<end>元素指定结束时间/日期，其值的一般形式在[ISO8601]中定义。

<!ELEMENT o-dd:interval (#PCDATA)>

在<interval>元素指定的时段内，可以对DRM内容行使权限，没有次数限制。在第一次行使相关权限时，<interval>开始计时。

<!ELEMENT o-dd:accumulated (#PCDATA)>

<accumulated>元素指定对DRM内容可以行使权限的最长累计时段。其值的一般形式在[ISO8601]中定义。在第一次行使相关权限时，<accumulated>开始计时。<accumulated>只在DRM内容使用时进行时间值累计。

<!ELEMENT o-dd:individual (o-ex:context)>

<individual>元素指定内容被绑定的个体。它由<context>子元素将内容和用户身份绑定。

<!ELEMENT oma-dd:system (o-ex:context+)>

<system>元素指定了DRM内容和权限对象可以被输出的目标系统，目标系统可以在强制的<context>元素中描述。

6. 继承模型（Inheritance Model）

用ODRL继承模型的有限子集，父权限对象可以为一个或多个DRM内容块指定许可和限制，每个DRM内容块由一个子权限对象控制。

<!ELEMENT o-ex:inherit (o-ex:context)>

为了允许定义父/子关系，<inherit>元素指定从一个权限对象到另一个权限对象的许可和约束的继承，父权限对象为DRM内容定义的许可和约束可以由子权限对象继承，通常由子权限对象引用DRM内容（如图5-12^[2]所示）。

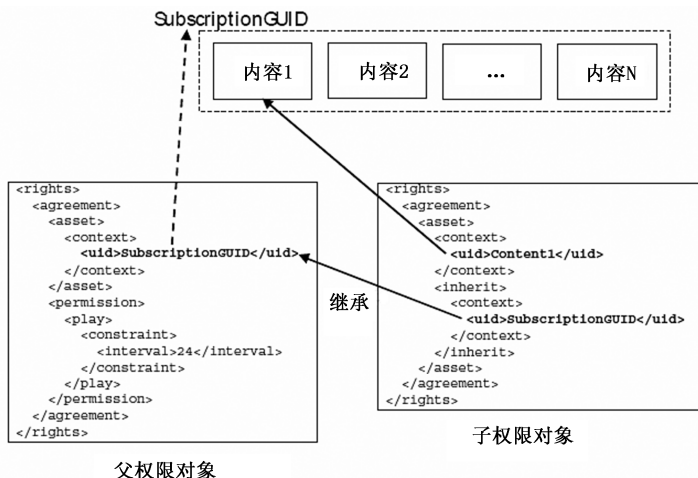


图 5-12 继承

7. 安全模型（Security Model）

安全组件是DRM系统的重要组成部分，OMA DRM 2.0提供了保密性、完整性和可认证性。

（1）权限对象 CEK 的保密性

内容的保密性是对DRM内容进行控制的基本部分，相应的XML安全元素描述下。

```
<!ELEMENT ds:KeyInfo (xenc:EncryptedKey?, ds:RetrievalMethod?)>
```

如果它包含在<asset>元素中，它包含<EncryptedKey>元素，对内容进行加密；如果它包含在<EncryptedKey>元素中，它包含<RetrievalMethod>元素，引用CEK的加密密钥REK（详见[1]）。

```
<!ELEMENT xenc:EncryptedKey (xenc:EncryptionMethod, ds:KeyInfo?, xenc:CipherData)>
```

<EncryptedKey>元素包含可选的<KeyInfo>元素、<EncryptionMethod>元素和<CipherData>元素。

```
<!ELEMENT xenc:EncryptionMethod (#PCDATA)>
```

<EncryptionMethod>元素是空的，它的Algorithm属性标识了用于CEK加密的算法。

```
<!ATTLIST xenc:EncryptionMethod Algorithm CDATA #FIXED
```

```
"http://www.w3.org/2001/04/xmlenc#kw-aes128">
```

Algorithm属性标识了用于CEK加密的算法为AES128，其加密形式在<CipherValue>元素中。

```
<!ELEMENT xenc:CipherData (xenc:CipherValue)>
```

<CipherData>元素包含<CipherValue>元素。

```
<!ELEMENT xenc:CipherValue (#PCDATA)>
```

<CipherValue>元素包含加密的CEK的base64编码值。

```
<!ELEMENT ds:RetrievalMethod (#PCDATA)>
```

<RetrievalMethod>元素通过其属性提供了对CEK加密密钥REK的引用。

```
<!ATTLIST ds:RetrievalMethod URI CDATA #REQUIRED>
```

URI属性提供了对CEK加密密钥REK的引用（详见[1]）。

(2) 与权限对象相关的 DRM 内容的完整性

DRM内容的Hash值包含在权限对象中。既然这个Hash值是签名的权限对象的一部分，那么当权限对象中的<uid>元素引用DRM内容时，它是没被篡改的，Hash函数的特性保证了内容的完整性。

<!ELEMENT o-ex:digest (ds:DigestMethod, ds:DigestValue)>

<digest>元素提供了与权限对象相关联的DRM内容的完整性， DRM内容由同一<asset>元素的<context>中的<uid>元素引用， <digest>元素包含<DigestMethod>元素和<DigestValue>元素。

<!ELEMENT ds:DigestMethod (#PCDATA)>

<DigestMethod>元素通过其属性标识计算摘要值的算法，这个摘要值包含在<DigestValue>中， <DigestMethod>元素本身是空的。

<!ATTLIST ds:DigestMethod Algorithm CDATA #FIXED
"http://www.w3.org/2000/09/xmldsig#sha1">

Algorithm属性标识摘要算法，它必须标识SHA-1算法。

<!ELEMENT ds:DigestValue (#PCDATA)>

<DigestValue>元素包含DRM内容的摘要值的base64编码，这个DRM内容由同一<asset>元素的<context>中的<uid>元素引用。

(3) 权限对象的完整性和可认证性

完整性保护防止非法修改为DRM内容指定的权限对象，包括但不限于对DRM内容添加、删除、修改的许可和约束，并且引用包含在权限对象中的DRM内容本身。

可认证性提供对原始权限对象的认证，它使DRM代理在接受权限对象前验证权限对象发行者的身份。

用来提供权限对象完整性和可认证性的功能在[1]中有详细说明。

5.6 OMA DRM 2.0 安全机制

OMA DRM 1.0 技术主要用于内容的保护，OMA DRM 2.0 技术在完全兼容 DRM 1.0 技术的基础上，对 DRM 规范做了大量改进，主要加强了通信和密钥的保护。它是针对功能更强大的终端设备设计的，这些终端设备拥有较多的内存和强大的处理能力，能播放高质量的音视频内容，能将受保护的内容发送给其他 DRM 设备或存储备份。

OMA DRM 2.0 的安全模式包括如下内容。

(1) 信任模式

OMA DRM 信任模式是基于公钥基础设施（PKI）的。若 RI 核实过 DRM 代理证书且该证书没有被吊销，则 RI 就信任 DRM 代理；若 DRM 代理核实过 RI 的证书，且该证书没有被吊销，则信任 RI。

(2) 机密性

通过对内容加密和对携带内容密钥（CEK）的 RO 加密绑定，确保了只有经过认证和授权的 DRM 代理才能访问受保护内容，未经授权方不能访问 DRM 内容。

(3) 认证

在 OMA DRM 的 4 通道注册协议、2 通道 ROAP 和 2 通道入域协议里,通过对实时或时间戳进行数字签名完成 RI 和 DRM 代理间的相互认证;在 1 通道 ROAP 里,通过对时间戳进行数字签名完成对 RI 的认证,但不能向 RI 认证 DRM 代理;在 2 通道离域协议里,通过对时间戳进行数字签名完成 RI 对 DRM 代理的认证,但不能向 DRM 代理认证 RI。

(4) 数据完整性保护

通过对 ROAP 和 RO 进行数字签名完成数据的完整性保护,防止对数据进行未经授权的修改。

(5) 密钥确认

通过受保护密钥和发送方 ID 上的消息访问控制(MAC)来完成密钥确认,受保护密钥的一部分作为 MAC 密钥。密钥确认确保能够接收含有受保护密钥的信息。

(6) 重放保护

时间戳提供了重放保护。

OMA DRM 2.0 的安全架构由 PKI 系统、权利对象获取协议(ROAP)、DRM 内容格式(DCF)、权利对象描述语言(REL)四个部分构成。各个部分在实现 DRM 系统应用的安全中起着不同的作用。公共密钥 PKI 系统作为安全基础设施平台是安全协议能有效实行的基础,一切基于身份验证的应用都需要 PKI 的支持。它可与 ROAP、TCP/IP、REL 相互结合实现身份认证、私钥签名等功能。基于数字证书和加密密钥,PKI 系统提供了一种在分布式网络中高度规模化、可管理的用户验证手段。

图 5-13 是 OMA DRM 系统的组成结构,包括了主要安全参与实体、网络安全协议平台和安全基础设施平台三个部分。网络安全协议平台包括 HTTP/WAP/MMS 协议。安全参与实体作为底层安全协议的实际应用者,相互之间的关系也由底层的安全协议和安全基础设施决定。当该安全构架运用于实际移动 DRM 管理中,这些安全参与实体间的关系,即体现为交易方(移动终端、内容发行者,权利对象发行者)和其他受信任方(无线认证中心)的关系^[6]。

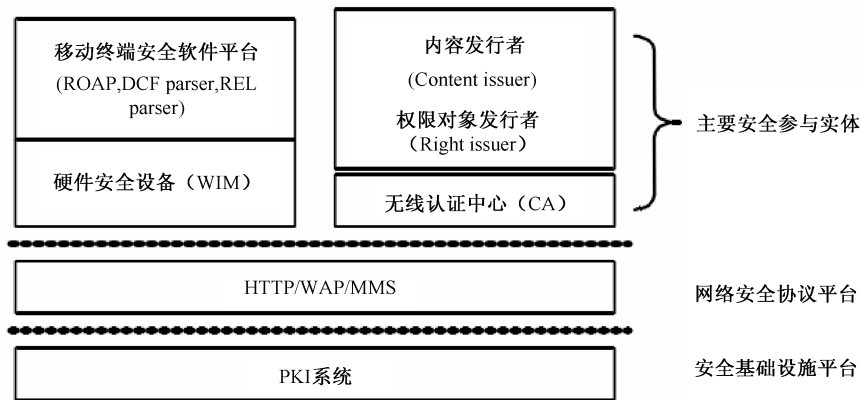


图 5-13 OMA DRM 系统的组成结构

ROAP 是 OMA DRM 2.0 中新增加并定义的关于安全获得权限对象的一整套安全协议。ROAP 是建立在 HTTP 之上的一个应用层协议,它采用 HTTP Accept Header 描述该报文的类

型（注册，获得 RO 或域管理）。ROAP 的所有报文格式采用 XML 拟写，充分利用了 XML 的优点，一方面具有很好的平台移植性，另一方面可以很方便地支持多权利对象的应用。ROAP 还可以与不同网络层次上的安全协议相互作用，比如 SSL/WTLS，为 DRM 系统提供更加坚固的安全系统。

该协议包含两个层次：一是在注册交互过程中，按照公钥交换体系，交换双方的公钥，为采用公钥加密传输版权对象做准备；二是简单请求应答协议设计，主要目的就是相关的版权对象下载到本地。

不同的 ROAP 子协议使用于不同的管理环境。下层基于 HTTP 与版权发行者进行通信，报文格式的描述采用 XML 的格式。在 OMA DRM 2.0 中，定义了 ROAP 后，不仅实现了版权对象的安全传输，而且还增加了对非连接设备和域的支持，扩充了 DRM 的商业应用范围。

虽然 OMA DRM 2.0 相对于 OMA DRM 1.0 在功能和安全方面有了很大的提高，但是 OMA DRM 2.0 也存在一些问题。

① 部署和实施难度较大。PKI 机制的引入，使 OMA DRM 2.0 的系统部署难度较大，如需建设 CA、管理和发放证书、建设数据库来维护证书信息等。

② 终端支持比较滞后。OMA DRM 2.0 中终端的实现涉及哈希摘要、数字签名和对 DRM 内容加密和解密等，因而对手机的运算能力要求非常高，硬件和软件资源占用非常大，耗电量大大增加，普通的 3G 终端难以实现。

③ 直播流业务的实现方式未确定。直播流媒体业务保护机制主要遵循 3GPP R6 标准对流媒体中的 RTP 包进行实时加密来实现。由于 OMA DRM 2.0 没有给出直播流业务的 DRM 实现机制的明确建议和实现方法，因此目前各厂商的 DRM 系统暂不支持 OMA DRM 2.0 直播流媒体业务的 DRM 服务器。

在 OMA DRM 2.0 体系中，采用分别发送方式发送受保护的 DCF 和 RO 是一个创新，能使内容适用于很多应用场景，因而得到了广大厂商支持。但是该体系目前尚存在以下不足。

① 由于受保护的 DCF 和版权对象被分开存放，当版权对象丢失或损坏时，用户就无法继续使用已付费内容，因而影响使用已购买的内容。

② OMA DRM 采取把 RO 和指定的一个或一组设备相绑定的机制，来确保只有授权用户才能访问内容和 RO。此机制给用户带来了很大不便：用户只能在特定的设备上使用已购买的 DRM 内容，而不能带到异地使用；由于 RO 和用户设备已绑定，用户不能升级自己的设备（如计算机）；用户不能将自己购买的 RO 转让给别人；攻击者可通过盗用用户计算机，非法使用数字产品。

5.7 AVS DRM 标准

5.7.1 AVS 标准概述

AVS 标准是中国数字音视频编解码技术标准工作组（Audio Video Coding Standard Workgroup of China）制定的《信息技术先进音视频编码》系列标准的简称，旨在制定一个具

有中国自主知识产权的, 有更高编码效率和性能的, 适用于数字电视、视频存储以及视频网络传输等不同码率应用的视频标准^[8]。

AVS 标准包括系统、视频、音频、数字版权管理四个主要技术标准和一致性测试等支撑标准, 其中的数字版权管理 (AVS DRM) 部分是为了适应数字电视广播、网络流媒体以及数字存储媒体等应用中对数字媒体版权管理的需要而制定的共性基础标准。AVS DRM 针对 IPTV 的实际应用环境提出了网络电视档, 它主要包括系统协议消息、媒体内容打包和权利描述语言等^[9]。

AVS DRM 包括三个档: 核心档、网络电视档、广播档。其中核心档定义可信解码器的技术要求, 广播档与网络电视档都建立在核心档基础上^[10]。

核心档定义可信解码器的构成及其各构成单元的功能和性能要求, 符合 GB/T 20090.6 的应用实现都必须支持核心档。广播档针对数字电视广播等单项广播应用, 基于 AVS 第一部分 (GB/T 20090.1) 所定义的传输流, 定义 AVS 接收终端能够解析的版权描述信息的语法。网络电视档针对 IP 传输环境下的音视频节目广播、点播和下载等服务, 基于 AVS 第八部分 (GB/T 20090.8) 所定义的网络封装格式和 AVS 第九部分 (GB/T 20090.9) 所定义的文件封装格式, 定义 AVS 接收终端能够解析的版权描述信息的语法和内容封装格式。此外, AVS DRM 支持采用国家密码管理局指定的密码算法。

AVS DRM 的概要参考模型如图 5-14 所示, 自内而外包括可信解码器、适配层和外围环境。

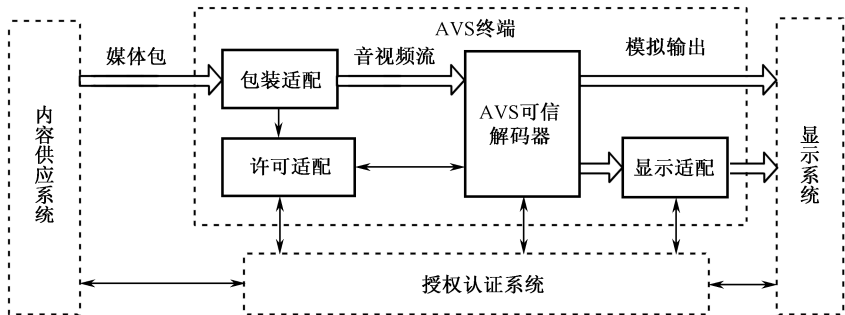


图 5-14 AVS DRM概要参考模型

可信解码器是普通解码器的扩展, 增加了认证、解密、明文重构等单元和输出加密等可选单元。可信解码器的外围环境包括内容供应系统、授权认证系统和显示系统, 标准的本部分定义可信解码器与这些系统之间的接口。适配层是可信解码器和外围环境的连接层, 解决可信解码器和外围环境之间的互连互操作问题, 包括: 适应内容供应系统的包装适配层、适应授权认证系统的许可适配层和适应显示系统的显示适配层, 其中显示适配层是可选的。

AVS DRM 对媒体内容加解扰采用对称加密算法, 对音视频的加扰可以采用全加扰或部分加扰的方式。身份认证采用公钥算法, 用于可信解码器的身份认证, 支持与其他设备建立安全信道, 支持数字签名。在密钥管理方面, AVS DRM 提出了内容密钥、业务密钥、认证公私钥的基本体系, 根据实际业务的需求, 可以对密钥体系进行扩展完善。

5.7.2 AVS DRM 核心档

AVS DRM 核心档主要定义可信解码器的构成及其各构成单元的功能和性能要求，是所有应用都必须支持的。

数字媒体版权管理最重要的一个目标是保护数字内容，应用系统的安全程度最终取决于系统中最薄弱的环节。在 DRM 中，最关键的环节是解码器，因为密钥管理、内容的解密、内容输出都在解码器环节进行，是最容易出问题的环节。因此，AVS DRM 采用可信解码器，在传统音视频解码器基础上增加了一些安全单元，如图 5-15 所示。

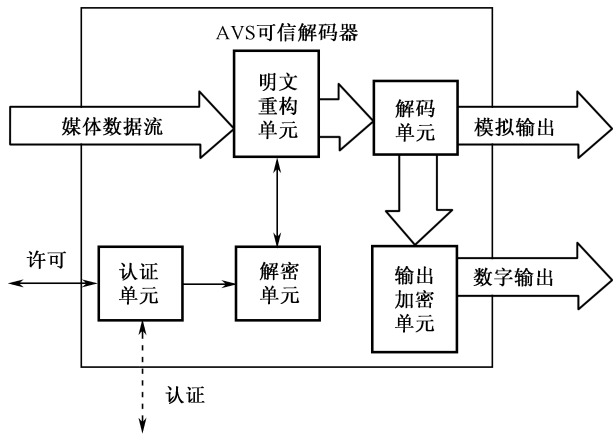


图 5-15 AVS可信解码器结构

认证单元是采用公钥算法的密码单元，用于可信解码器的身份认证、许可认证等，支持与其他设备建立安全信道，支持数字签名及签名验证。解密单元对加密过的 AVS 内容进行解密。明文重构单元利用解密后的数据，把原始加密数据的媒体流恢复成符合 AVS 标准的明文媒体流，从而使得重构结果能够直接送入符合 AVS 标准的解码器。可信解码器还可包含解码输出流加密单元，支持把解码后的音频、视频流以加密方式输出给显示设备。

AVS 可信解码器把认证单元放入解码器。认证单元可以和授权认证系统建立安全通道，用于传送内容密钥等秘密信息，然后直接交给解密单元，这种一体化设计保证了解码器这个关键环节的安全。AVS 可信解码器可以和智能卡配合使用（实现权限管理和密钥管理），以提供管理方便性。AVS 可信解码器也可以直接和内容运营商的前端授权认证系统建立安全通道，这种方案成本更低。

AVS 可信解码器不仅可用于数字电视这种广播环境，也可以用于网络电视这种交互式应用，能够作为三网融合条件下的统一音视频终端。AVS 可信解码器采用了国家密码主管部门指定的安全算法，实现了安全模块的统一，可解决目前数字电视领域不同 CA 造成的市场割裂问题，也能够防止 DRM 分割数字媒体市场的潜在风险。

5.7.3 AVS DRM 权利描述

AVS DRM 数字权利描述语言（AVS DREL）为数字版权管理提供了一套统一的权利描述

机制,实现对实体间许可协议的描述,为在开放和可信任环境中进行授权管理和访问控制提供了一整套基于许可协议的语法和语义规则。该标准将数字内容的权利信息包装成机器可读的许可证。借助许可证,DRM 系统可规范化发布、传输和认证数字资源的权利。数字权利描述语言部分中所使用的基于 XML 的许可证以及在 AVS DRM 系统协议机制中所使用的基于 XML 的消息,推荐使用二进制的信息格式以减少 AVS 解码器所需的负荷。

AVS DRM 许可证由以下三部分组成^[1]。

① 权利发布者:描述权利发布者的基本信息,许可证必须包括权利发布者。

② 许可单元:许可证的基本单元。许可单元在许可证中是必须出现的,但是出现的个数没有限制。在同一个许可证下的许可单元必须具有相同的权利接受者。许可单元针对每个资源向权利接受者发布某种权利,依次包含主体、权利、资源、约束和义务 5 种元素。许可单元与权利接受者、权利和资源是一一对应的,但是许可单元可以对定义的权利设置多个约束条件和义务条件。

③ 许可证签名:用于保证许可证的完整性和不可否认性。

AVS DRM 许可证基本结构如图 5-16 所示。

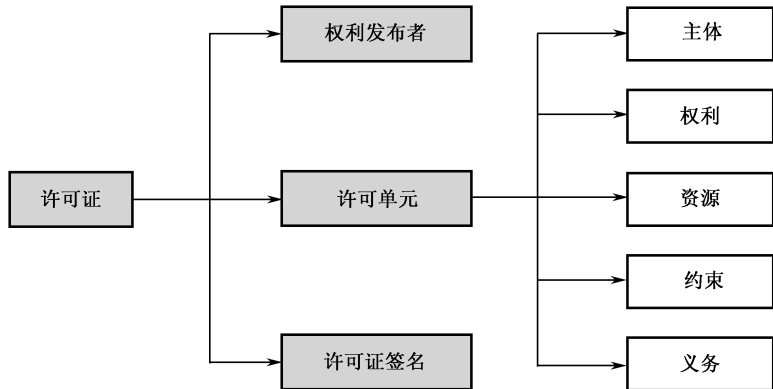


图 5-16 AVS DRM 许可证基本结构

AVS DREL 分为三个部分,分别是信息模型、数据字典和 XML 语法。信息模型对数字权利描述语言的语法结构和框架模型进行描述;数据字典给出了 AVS DREL 语法框架下的元素语义定义;XML 语法作为附录,描述了如何以 XML 文档的形式对数字权利进行语法描述,通过 XML Schema 即可对 XML 文档的有效性进行验证。

AVS DREL 基于核心档所定义的密码算法和安全协议,实现身份认证,维护内容机密性和完整性,支持隐私保护。数字版权管理的协议消息需要涉及以下两大类协议消息:注册管理(认证媒体终端身份)和许可证(管理媒体许可)获取/撤销管理。AVS DRM 网络电视档定义了双向通信网络环境下的媒体终端和注册服务器、许可服务器之间的协议和消息格式,并假定通信双方均支持公钥体系来辅助双方之间的相关事务。

5.7.4 AVS DRM 网络电视档

IPTV 是基于宽带互联网的一种以数字音视频资源为主体,以电视机、计算机等为显示终

端的媒体服务，是 Internet 业务和传统电视业务融合后产生的新业务。IPTV 是基于内容管理的、开放的、交互的音视频内容和业务经营系统，由于 IP 网络上数字化的节目内容播发过程中存在许多安全隐患，因此，有效的版权管理能够实现音视频节目的版权保护和合法消费。保护 IPTV 数字媒体内容版权的安全，需要建立起一套包括加密、认证和权限管理的安全机制，通过采用媒体内容加密、身份认证、颁发用户权利许可证等安全手段，使得只有授权的用户才能消费特定的节目，只有允许的节目才能播出，防止非法收看、传播或篡改。

DRM 技术从技术上防止数字内容的非法使用或复制，最终使得用户必须得到授权才能够使用数字内容。

AVS DRM 针对 IPTV 提出了网络电视档 DRM 参考模型（图 5-17），它是 AVS DRM 概要参考模型的一种具体化和扩展。网络电视档充分考虑了 IPTV 的平台特点及基本业务需求，其技术要求包括：系统协议消息、媒体内容打包（包括文件格式和 RTP 封装）和权利描述语言。

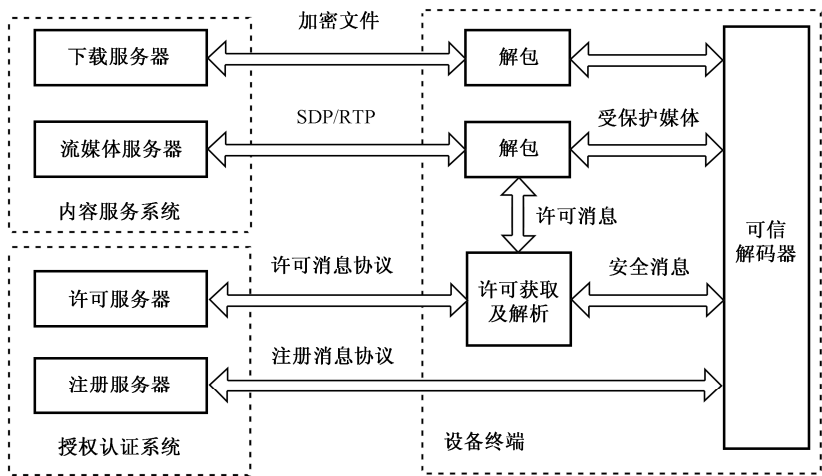


图 5-17 网络电视档DRM参考模型

媒体内容打包格式针对 IPTV 应用中的流服务和文件服务两种形态，定义了内容被保护情况下数字媒体文件的封装格式和在 IP 网络上传输时的 RTP 打包格式，即 RTP 净载动态保护格式和支持版权管理的 AVS 文件格式。RTP 净载动态保护格式是指流媒体服务器在实时发送已打包好的明文 RTP 包时，对其净载部分进行动态选择性加密的传输保护机制，在流媒体的传输过程中，流媒体服务器可以动态地更换密钥，增强对流媒体内容的保护强度。支持版权管理的 AVS 文件格式是对 AVS 第九部分的扩展，其内容的加密以 NAL 单元为单位。

网络电视档采用 AVS DREL 数字权利描述语言，其中对权利的限制推荐如下^[11]。

- ① 普通权利的使用类型中支持：输出、播放。
- ② 重用权利中支持：修改、分割、打包。
- ③ 管理类型中支持：移动、复制、备份、保存。
- ④ 高级权利中支持：转让类型和撤销类型。
- ⑤ 时间的约束中支持：次数、累加时间、时间段、使用周期。

参考文献

- [1] Open Mobile Alliance. DRM Architecture, 2008.
<http://www.openmobilealliance.com>
- [2] Open Mobile Alliance. DRM Rights Expression Language, 2008.
<http://www.openmobilealliance.com>
- [3] Open Mobile Alliance. DRM Content Format, 2008.
<http://www.openmobilealliance.com>
- [4] Open Mobile Alliance. DRM Specification, 2008.
<http://www.openmobilealliance.com>
- [5] 数字音视频编解码技术标准工作组.
<http://www.avs.org.cn/faq.asp>
- [6] 肖利. OMA 数字版权管理的研究与实现. 电子科技大学硕士学位论文, 2007.
- [7] 魏景芝, 杨义先, 钮心忻. OMA DRM 技术体系研究综述. 电子与信息学报, 2008, 30(3): 746-751.
- [8] 叶松, 于志强, 唐凌, 等. AVS DRM 标准在 IPTV 中的应用研究. 现代电子技术, 2010, (3): 40-43.
- [9] 叶松, 等. AVS DRM 标准在 IPTV 中的应用研究. 现代电子技术, 2010, (3): 40-43.
- [10] AVS 工作组. GB/T20090.6-YYYY 信息技术, 先进音视频编码 (第六部分) 数字媒体版权管理
- [11] 黄铁军. AVS 数字媒体版权管理标准. 中国数字电视, 2007, 6(34): 46-49.

权利描述语言

权利描述是指描述数字内容的授权信息，权利描述语言（Rights Expression Language, REL）用于构造许可证，描述数字内容或服务的使用权限，即描述用户对资源拥有的使用权限，比如说明该 DRM 内容可以被预览。REL 能够表达不同的许可和约束，并提供了一种简明的机制表达 DRM 内容的权限对象。DRM 内容消费需要根据其权限对象的规定、许可和约束，这些都记录在权限对象中，而不是 DRM 内容中。使用 REL 定义的权限对象保证只有被认证的设备才能使用媒体内容。

REL 是 DRM 领域的重要研究课题，也是 DRM 技术体系的关键与核心部分，当今 DRM 技术标准的争夺也主要体现在权利描述语言标准的争夺上。REL 必须方便易用，具备开放性、灵活性、可扩展性和可机读性，支持各种数字内容各类使用权利的描述。目前存在两类 REL，一类是基于逻辑的，以 LicenseScript^{[1][2]}为代表，尚处于理论研究阶段；另一类是基于 XML 的，已经处于实用阶段。

XrML（Extensible Right Markup Language，可扩展权利标记语言）^{[3][4]}和 ODRL（Open Digital Rights Language，开放数字权利语言）^[5]是当前发展最为完善的两个基于 XML 的权利描述语言，已分别被有关标准组织采纳。2000 年 4 月，ContentGuard 公司发布了 XrML 1.0 及相关的 XrML SDK 体系，为描述和实现数字版权管理提供了完整的解决方案，2002 年发布 XrML 2.0。2001 年 11 月作为对 ISO 下的 MPEG 所发布的“权利数据字典和权利描述语言提案征集”的回应，ContentGuard 公司将 XrML 2.0 提交给了 MPEG，2002 年 5 月被 MPEG-21 工作组 REL 研究小组采用，作为制定 MPEG-21 REL 的工作基础，并在 2002 年 3 月至 2003 年 7 月间，对 XrML 进行了标准化，2003 年底 MPEG-21 REL 成为标准，2004 年 3 月正式出版，国际标准编号为 ISO/IEC 21000-5:2004。目前 XrML 已成为了 DRM 实质上的标准。

ODRL 的研究可追溯到 1997 年，以自动化权利保护为研究专题的 John S. Erickson 从 Dartmouth 大学毕业后，在 HP 实验室继续对这块领域进行研究，2000 年提出 ODRL 语言。2002 年，ODRL 发展出新的版本为 ODRL 1.1，2003 年被纳入成为 W3C 标准，2005 年被 NISO（National Information Standard Organization，美国信息标准化组织）纳入成为 NISO 标准。

但 REL 对开放性、灵活性、可扩展性以及支持各类使用权限描述的要求使基于 XML 的语言难于满足要求。逻辑语言由于在表达力、灵活性和语义完整性上的优势，基于逻辑的 REL

语言的研究逐步受到重视。

6.1 XrML 的数据模型

权利描述语言在数字版权管理中起着很重要的作用，它用来描述主体、权限、资源和条件及其相互之间的关系。XrML 是一种成熟并广泛使用的权限描述语言，XrML 2.0 采用一种简单的、可扩展的数据模型来定义这些关键的概念和元素。

6.1.1 数据模型中的实体

XrML 数据模型由四个基础实体构成，即主体 (Principal)、权限 (Right)、资源 (Resource) 和条件 (Condition)。

1. 主体

主体标识着资源使用者的身份。在数字版权管理系统中，通常将主体标识信息和权限、资源绑定起来，通过数字证书和签名验证机制，以证实它的唯一性，即特定的资源、特定的权限在特定的条件下，主体只能是唯一的。这样才能保证满足权限和条件的唯一主体使用相应的资源。对于主体来说，它的身份必须是可信任的，用主体拥有的公/私钥对中的私钥作为验证过程中的密钥，来验证它的身份。

2. 权限

权限是被授权的主体在一定条件下使用相应资源的权利的集合，具体指使用资源时的一个操作或一系列操作。XrML 2.0 提供了一个抽象的<right>元素节点来描述权限，它包含各种具体的操作，如播放、打印等。

3. 资源

资源是主体被授权使用的对象，它可能是一种数字化的产品资源、一种服务或主体可以拥有的信息。XrML 2.0 定义了一个抽象的<resource>元素节点，基于这个元素，它又定义了许多扩展节点来应用于具体的资源上。

4. 条件

条件是权限执行时需要满足的因素集。简单条件可能要满足权限执行时的时钟要求，而复杂条件则包含权限执行、条件验证和平台移植时需要满足的可扩展的需求集合。XrML 2.0 定义了一个抽象的<condition>元素节点，基于此节点，它定义了一些具体的标准化的条件。

6.1.2 实体之间的关系

将 XrML 2.0 数据模型的四个基础实体主体 (Principal)、权限 (Right)、资源 (Resource) 和条件 (Condition) 联系在一起的是授权 (Grant)，它是一个对数字权利进行描述的最基本的、完整的结构 (图 6-1) [6]。

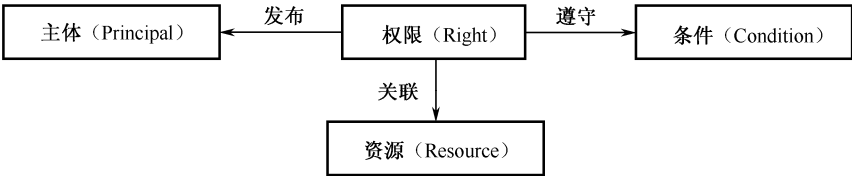


图 6-1 数字权利描述的基本结构

从图 6-1 可看出，主体是授权的对象；权限是使用权利的集合，它与资源紧密相关；资源是权限使用的对象，是版权保护的目标；条件是权限执行需要满足的软硬件因素的集合。

XrML 定义了一个 <grant> 元素节点，其构成如图 6-2 所示 [3]。

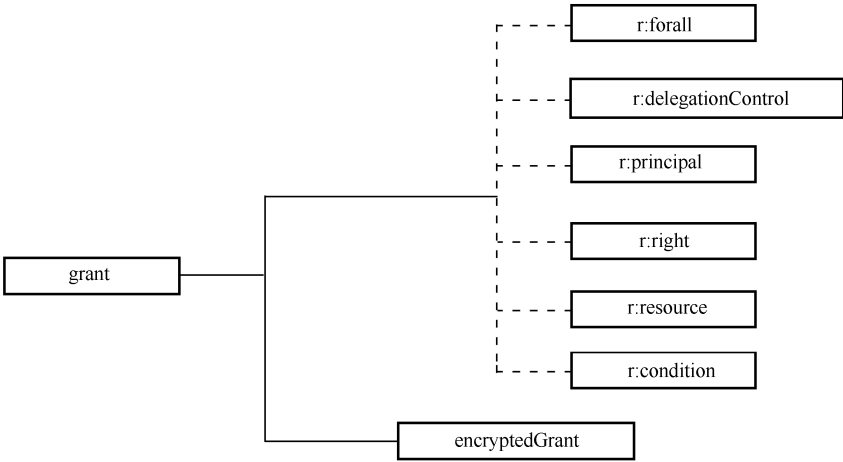


图 6-2 grant 元素节点的构成

在图 6-2 中，<encryptedGrant> 元素节点存储对 <grant> 元素加密后的数据；<r:forall> 元素表示资源的主体为所有人的情况，一般很少用；<r:delegationControl> 元素表示由第三方代理或委托时的元素节点，也不常用到；而 <r:principal>、<r:right>、<r:resource>、<r:condition> 4 个元素节点分别代表 XrML 的 4 个基础实体，在具体的 XrML 应用中经常用到。

在一个系统环境中，需要确定一些附加的信息，如发布授权的主体标识和许可证标识。出于这个原因，XrML 定义了许可证 (license) 结构，每个 license 包含一个或多个 grant，可以认为是一组授权。

在 XrML 2.0 中，license 是一个关键的顶级结构。从概念上讲，license 是 grant 的容器。许可证的基本结构 (图 6-3) 包括以下内容 [3]：

- ① 一个授权（grant）的集合：向主体授予在某种条件下对特定资源所享有的权限。
- ② 发行者（issuer）：对颁发许可证的主体进行身份标识。
- ③ 各种附加信息：对许可证进行描述。

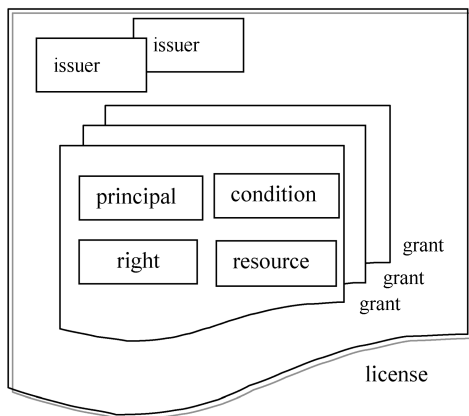


图 6-3 XrML 许可证结构

许可证可由颁发它的主体进行签名，表示颁发者确实授予了包含在许可证中的权限。在语法上，多个颁发者可以对同一张许可证签名。然而，不存在与联合签名有关的语义，所以看起来像是每个颁发者单独地对许可证的副本进行签名。

在 XrML 中，定义了一个 `<license>` 元素节点，它是一个非常重要的顶级元素节点，它代表着一个未加密的数字证书的开始。`license` 和 `certificate` 在数字版权管理中有一定的区别，`license` 往往表示一个元素节点或一个证书申请行为，而 `certificate` 则代表着一个用户、组织和服务器等的数字证书文件，后者具有合法性和权威性。`<license>` 元素节点的构成如图 6-4 所示^[3]。

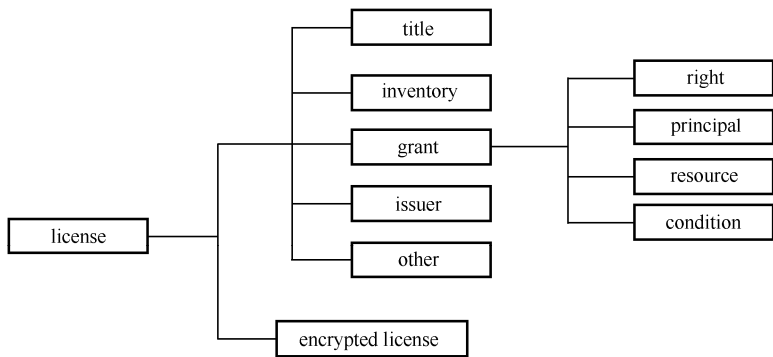


图 6-4 license 元素节点的构成

从图 6-4 可以看出 `<license>` 元素的构成。其中，`<title>` 元素表示证书的颁发机构名称；`<inventory>` 元素主要存放一些记录信息，不常用；`<issuer>` 元素常常存放的是签名算法、签名值和摘要信息；`<other>` 元素表示其他信息，可以扩展。另外，还附带一个 `<encrypted license>` 元素节点，存储元素数据加密之后的值。

`licenseGroup` 是 `license` 的容器，使用 XrML 定义的 `<licenseGroup>` 元素节点，可以实现一个 XrML 证书文件中定义多个 `license`。在某些情况下，可能存在多个用户使用同一个资源，

并且使用资源的权限和条件相同，使用 licenseGroup 就可以实现证书发布方在同一个 XrML 证书文件中进行相互独立的数字签名和认证，从而方便于证书管理。

6.2 数据模型在 XML Schema 中的封装

6.2.1 XrML 的组织结构

XrML 2.0 采用 W3C 的 XML Schema 技术进行描述和定义。XrML 提供了一个框架，可以在整个工作流或资源生命周期的各个阶段中表达不同的权限，此外它还定义了一些与资源格式以及商业模型无关的词汇（大约 100 个），利用这些词汇可以给任何形式的数字化资源定义权限，另外通过语法规则和唯一解释语言的处理规则以保证语言的精确性，最后利用 XML Schema 的扩展机制在核心结构的基础上可实现其扩展性。使用 XrML，数据资源的拥有者和发布者可以指定允许使用该数字资源的主体，指定这些主体能够获取的权限，以及这些权限使用的条件和属性。

XrML 2.0 的主要设计目标是在 XrML 核心结构不进行实质改变的前提下，支持可扩展性。XML Schema 的扩展机制使 XrML 2.0 在它的表示和扩展上能提供高度丰富性和灵活性。XrML 2.0 大体可以分为三部分（图 6-5）^[3]。

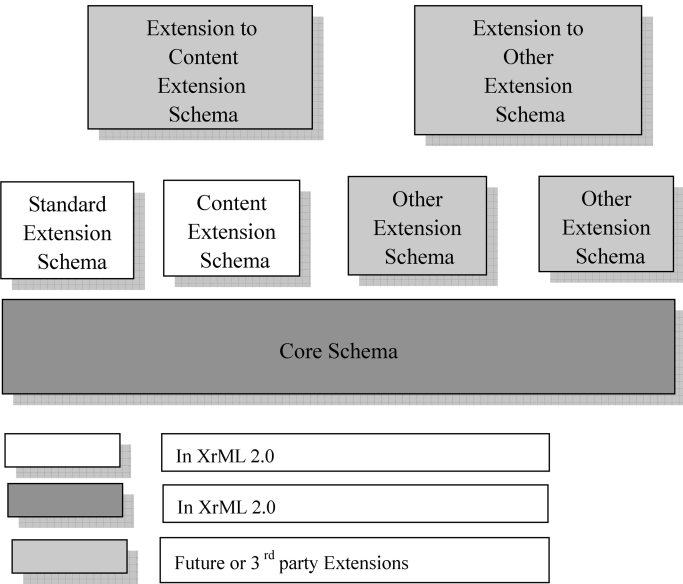


图 6-5 XrML 组织结构

1. 核心模式（Core Schema）

核心模式主要定义了 XrML 2.0 语义的基本结构，特别是那些用来衡量信赖决策的概念定义。

2. 标准扩展模式 (Standard Extension Schema)

标准扩展模式进一步扩展了经常使用到的设定描述 (如 `payment` 和 `name`)，但不一定使用 XrML 的语义。

3. 内容扩展模式 (Content Extension Schema)

针对 `rights`、`conditions` 和版权管理的定义加以扩展，特别是与数字产品 (如电子书、音频、视频) 相关的版权管理的概念。

XrML 2.0 的可扩展模式为开发者提供了强大而灵活的权利描述功能。

6.2.2 强制项和可选项

实际上，在 XrML 中大多数定义是可选的，但为了生成有效的许可证，某些元素是强制定义的，如 “`license`” 至少要包含一个 “`grant`”。强制项和可选项的使用允许创建简单或复杂的表示，可以在需要的时候使用可选项。

强制项集合的元素包括：

- ① 许可证 `license`；
- ② 授权 `grant` 或授权组 `grantGroup`；
- ③ 权限 `right` 或它的一个替代。

权限 `right` 是抽象的，它将由 `issue`、`obtain`、`possessProperty`、`revoke` 或在扩展中定义的权限替代 (如在内容扩展中定义的 `play` 权限)。

定义许可证 `license` 的所有强制项是内核结构的一部分。在标准和内容扩展中的所有定义都是可选的。

最简单、最小的许可证可以用强制项和一对可选项来构造。

例如，下面的许可证说明密码的拥有者为 Alice Richardson。

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- This is a simple certificate -->
<license xmlns="http://www.xrml.org/schema/2001/11/xrml2core"
  xmlns:sx="http://www.xrml.org/schema/2001/11/xrml2sx"
  xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.xrml.org/schema/2001/11/xrml2cx..\schemas\xrml2cx.xsd">
  <!-- Certify that the following key holder has the common name "Alice Richardson"-->
  <grant>
    <keyHolder>
      <info>
        <dsig:KeyValue>
          <dsig:RSAKeyValue>
```

```

        <dsig:Modulus>Fa7wo6NYfmvGqy4ACSWcNmuQfbejSZx
        7aCiblgkYswUeTCrmS0h27GJrA15SS7TYZzSfaS0xR9lZ
        dUEF0ThO4w==</dsig:Modulus>
        <dsig:Exponent>AQABAA==</dsig:Exponent>
    </dsig:RSAKeyValue>
</dsig:KeyValue>
</info>
</keyHolder>
<possessProperty />
<sx:commonName>Alice Richardson</sx:commonName>
</grant>
</license>

```

如前所述，数字版权管理的基本前提是建立一个实体，大多数权限的表示依赖于产业类型。上例没有发布者 issuer，尽管如此，issuer 在大多数应用中都是需要的。这个例子是一个有效的 XrML 许可证。

6.2.3 核心模式

XrML 2.0 最主要的部分就是核心模式（表 6-1^[3]）。核心模式中的元素和类型定义了结构化的有效性语义，它们是 XrML 2.0 规范的基础。XrML 2.0 内核概念包括 license、grant、principal、right、resource 和 condition，并且用抽象的方式把后面四个定义为元素，对 XrML 核心的扩展可以将这些元素定义成特殊的格式，并应用到具体的系统中。

表 6-1 核心模式一览

父 元 素	元 素	定 义
license	—	grant 的容器，由 issuer 发布
	title	对 license 的描述
	grant grantGroup	grant 和 grantGroup 元素包含在 license 中，用来传递授权策略
	issuer	在 license 中的 issuer 元素包含两个信息：① 发布 license 的详细信息（日期，撤销机制等）；② license 的数字签名
	inventory	提供一个类似宏的语法机制，减少 license 中的冗余
	(any)	采用 XML Schema 的通配符结构，license 提供了一个可扩展的方式，使 license 的发布者可以恰当而方便地添加附加内容，例如，与认证和授权相关的信息，但这不是 XrML 2.0 内核结构的一部分
	encryptedLicense	提供 license 的内容加密机制
grant	—	grant 说明在一定的条件(condition)下，某个主体(principal)对某个资源(resource)具有某种权限(right)，这个结构是 XrML 2.0 权限管理和授权策略语义的核心
	forAll	forAll 元素定义了一个集合变量，可以由 grant 中的任何元素所使用
	principal	principal 元素是抽象的，使用时通常用一个元素来替代，例如“keyholder”或“allPrincipals”

续表

父 元 素	元 素	定 义
	Right	Right 元素是抽象的，使用时通常用一个元素来替代，例如，在内容扩展中定义的权限“play”、“print”、“copy”和“edit”
	resource	resource 元素是抽象的，使用时通常用一个元素来替代，例如 XrML 核心中定义的“digitalResource”
	condition	
	delegationControl	在环境控制策略下，grant 可以由 delegationControl 元素表达的语义指定被授权者（例如，一个目标主体）
	encryptedGrant	encryptedGrant 指定使用 XML 的加密语法和处理标准对 grant 加密
grantGroup	—	和 grant 类似，grantGroup 可以和 grant 的所有子元素相关联，也可包含其他 grant 或 grantGroup。当几个 grant 整体发布，又有公共元素时使用
principal	allPrincipal	allPrincipal 元素包含一个 principal 的集合，如同整体被标识的实体一样共同完成操作
	keyholder	表示一个主体 principal，由其所拥有的某个密钥所标识，如 PKI 环境中的私钥
right	issue	issue 元素可在多层模型中使用，分发者被授权发布某些权限，也可用于点对点的分发，一个实体可向另一个实体分发具有特定权限的许可证
	revoke	当一个主体对某个权限执行 revoke 操作时，将撤销由数字签名所给予的授权
	possessProperty	possessProperty 元素授予主体声称拥有某些资源的权限
	obtain	obtain 元素表示获取 right 的权限
resource	digitalResource	可由元素来确定数字内容，也可由外部文件或 Web 站点来确定
condition	allConditions	当 allConditions 中的每一个条件都满足时，allConditions 条件满足
	validityInterval	表示相邻的、不间断的时间间隔
	revocationFreshness	表示时间间隔的上界。XrML 强制执行机制必须据此检查认证签名的有效性（检查签名是否被撤销）
	existRight	指定了相关权限得以执行所必须具备的另一项权限，只要该权限存在，不管是否有效
	prerequisiteRight	指定了相关权限得以执行所必须具备的另一项权限，不仅要求该权限存在，而且要在有效期内
其他的核心类型和元素		
aAny	trustedPrincipal	trustedPrincipal 元素指定了一个策略，根据这个策略，主体在某些场合是可信的
any	serviceReference	用于指定 Web 服务
N/A	licenseGroup	许可证的容器

6.2.4 标准扩展模式

标准扩展模式定义了对内核模式的扩展（表 6-2^[3]）。特别地，它用附加条目的方式扩展了 XrML 核心中的条件（Condition）类型，用来支持需要执行一个权限操作（如使用跟踪）的外部服务、支付条件和方法，以及时间条件等。

表 6-2 标准扩展一览

父 元 素	元 素	定 义
grant	statefulCondition	某些条件可能和外部授权数据块绑定。在执行权限操作时，需要用这个元素查询或处理外部数据块的值
statefulCondition	stateReference	表示状态查询或处理的方法，它的类型是在 XrML 核心中定义的 serviceReference
grant	stateReference-valuePattern	用于指定状态信息的值
	exerciseLimit	表示相关权限的执行次数（例如，一首歌曲的播放次数）
	seekApproval	执行相关权限前必须与特定服务联系并取得其认可
	trackReport	权限的执行必须由指定的跟踪服务监控等
	trackQuery	追踪由 trackReport 更新的状态
	validityIntervalFloating	权限执行的时间长度
	validityTimeMetered	权限执行的累计时间
	validityTimePeriod	权限执行的时间周期
	fee	表示与付费方式相关的条件，包含以下子元素：paymentAbstract、min（应支付的最小量）、max（应支付的最大量）和 to（付费给谁）
fee	paymentAbstract	这个抽象元素由 payment 扩展来替代，以表示更明确的支付形式
grant	territory	指定权限执行的场合，可以是实际的区域，也可以是虚拟的位置，子元素表示位置（/region、/county、/state、/city、/postalCode、/street）和域（/url）
payment 扩展		
fee	paymentAbstract	这个抽象元素由下面描述更明确的支付形式替代
	cash	支付所包含的类型
	paymentFlat	表示固定费用。子元素是 rate（总额）和 paymentRecord（如果支付发生，则表示支付的值）
	paymentMetered	指定根据使用的持续时间支付费用。例如，一个人玩游戏，根据玩游戏的时间长度支付费用。子元素是 rate（每段时间的总额）、per（时间段）、by（用于计算总额的周期量），以及 phase（宽限期/直到下一次计算）
	paymentPerInterval	指定在已付费的时间期限内可操作的权限。例如，一个人购买了玩游戏时间段，他可以一直游戏直到时间用完。子元素为 rate（每段时间的总额）、per（时间段）、以及 paidThrough（已付费的时间）
	paymentPerUse	指定可操作的权限，每次使用都要付费。例如，支付了相应的费用，才具有听音乐的权限。子元素为 rate（每次使用或使用次数的总额）和 allowPrePay（指定剩余使用次数和/或初始的使用次数）
	bestPriceUnder	表示账目结算时，付费可以是动态的，也可以是确定的，它用于特殊的优惠、折扣、定价，在执行操作权限时，其信息不提供给可信存储库，购买授权之前无需与经销商通信
	callForPrice	callForPrice 和 bestPriceUnder 很相似，适用于价格动态变化的情况。然而，和 bestPriceUnder 不一样，购买授权前，需要和发行商沟通确定价格；如果可信存储库不能与发行商通信，则交易不能完成
	markup	表示按其他费用的百分比计算的费用。例如，购买一个数字作品的拷贝，发行商可能想增加 10% 的费用；或政府想征收数字作品的销售税
name 扩展：possessProperty、资源名称（name）和它的扩展一起，使许可证可直接表示与名称相关的授权		

续表

父 元 素	元 素	定 义
需要 resource 的任何地方	emailName	通常包含一个字符串，指定互联网上的 E-mail 地址（根据 RFC822/2822）。许可证用权限 possessProperty 可将元素 emailName 与主体关联
	dnsName	通常包含一个字符串，指定域名。许可证用权限 possessProperty 可将元素 dnsName 与主体关联
	commonName	通常包含一个字符串，指定别名。许可证用权限 possessProperty 将元素 commonName 与主体关联
	x509SubjectName	通常包含一个字符串，从 X509 证书中指定主题名称。许可证用权限 possessProperty 将元素 x509SubjectName 与主体关联
	x509SubjectName- Pattern	和 x509SubjectName 匹配的模式
对 revocation 的扩展		
	revocable	和权限 revoke 一起使用，revocable 元素由它的文本值或间接表示（如它加密的摘要值）识别（被撤销的）签名

6.2.5 内容扩展模式

XrML 内容扩展模式扩充了核心模式，描述了数字作品的权限、条件和元数据（表 6-3^[3]）。特别地，定义了传播和使用数字内容的特定权限，这些权限包括以下几种。

- ① 呈现权（Render Rights），管理数字作品的表现。包括三种权利：演播（play），呈现作品内容的暂时形式（如展示一本书、播放一段视频或音频、玩计算机游戏等）；打印（print），是指在原有库（repository）之外制作作品的永久副本（如复印一本书、保存图片到可移动硬盘、在磁带上录音等）；输出（export），是指制作作品的数字拷贝但没有附加相应的权利和条件。
- ② 传送权（Transport Rights），管理数字作品在不同的库（repository）之间的移动。包括三种权利：复制（copy），制作作品的副本；转移（transfer），将作品转移到其他库并从原有库中删除被保护内容；借出（loan），将作品借出一段时间（原来的那份不能再使用）。
- ③ 作品派生权（Derivative Work Rights），管理数字作品的整体或部分复用，或利用数字作品创建或组合新的作品。包括三种权利：编辑（edit），改变原作品以创建新作品；摘录（extract），使用作品的一部分以创建或组合新的作品；嵌入（embed），将作品副本嵌入组合作品。
- ④ 配置权（Configuration Rights），管理库中系统软件的添加/删除。包括两种权利：安装（install），使软件能在库中运行（如检查软件是否经鉴定、是否被篡改、是否与库兼容等）；卸载（uninstall），软件不再运行并恢复到安装前的状态（但并不从库中删除程序文件）。
- ⑤ 文件管理权（File Management Rights），管理两种操作：获取库的目录信息以方便库与库之间的通信（如执行转移或借出权），制作备份和由备份恢复数字作品。包括 9 种权利：读取（read），从库中读取作品；写入（write），向库中写入作品；执行（execute），在库中执行作品；删除（delete），从库中删除作品；备份（backup），为作品创建备份（备份是加密的，恢复之前不能再使用）；恢复（restore），在受控方式下将备份转换为可用作品；验证（verify），

检查库中作品的真实性；目录管理（manageFolder），创建、命名子目录和在目录之间移动文件、子目录；获取目录信息（accessFolderInfo），获得库中目录内的作品信息。

表 6-3 内容扩展一览

类 型	元 素	定 义
对权限的扩展：替代抽象类型 right		
文件管理权	accessFolderInfo	表示传递或显示作品所在目录信息的权限
	backup	表示为作品创建备份的权限，防止因意外读取失败
	delete	表示在一个库中删除作品的权限。很多人可以登录固定库，出于偶然或恶意删除文件，所以必须控制删除权限
	execute	表示在一个库中对一个资源操作的权限
	manageFolder	表示完成下列操作的权限：创建子目录、命名子目录，重新配置子目录（如移动目录中的文件和目录）
	read	表示在库中读取作品的权限
	restore	表示从一个备份中恢复一个作品的权限，将备份转成可用的形式
	verify	表示在一个库中检测作品真实性的权限。身份鉴别包括采用公钥/私钥鉴别数据的签名。完整性验证保证接收的数据和发送的数据匹配，它保证数据没有被篡改过
	write	表示写或保存一个库中作品的权限
传送权	copy	表示创建一个新的数字作品拷贝的权限
	loan	表示在一个特定的时间内，将一个作品借给另一个主体的权限。当作品借出后，原始拷贝不能使用
	transfer	表示将作品传送到另一个库的权限，将作品从原始的位置移除
作品派生权	edit	表示在原始作品的基础上对作品进行修改，以创建一个新作品的权限。edit 和 extract 相似，它创建一个新作品。和 extract 不同的是，它赋予对作品进行修改的权限
	embed	表示将作品作为组合作品的一部分的权限。一个 embed 操作将一个作品的拷贝嵌入组作品
	extract	表示使用作品的一部分去创建一个新作品的权限。被提取的材料被创建成一个新作品，和原始作品分离。extract 和 edit 不同，没有授予修改作品的权限
呈现权	export	表示将作品的原始拷贝从源库输出的权限。例如，以明文保存一个受保护的（加密的）作品。export 不同于 copy，export 是传送到不安全的明文形式的库中，而 copy 是传送到另一个安全的库中
	play	表示以暂态形式呈现作品的权限，以适当的内容类型（例如显示一本书、播放一个声音片段或放映一段视频）
	print	表示在库的控制之外，以固定的、非数字化的形式呈现作品拷贝的权限。这个权限操作可能是打印一本书的硬拷贝或在一个磁带上创建一个视频记录
配置权	install	表示在库中将软件制成可执行形式（安装）的权限，包括下列的一些操作，例如，检测软件是否被篡改，和库是否兼容
	uninstall	表示使软件不能运行的权限。将它恢复到安装前的状态
对 resource 的扩展		
	digitalWork	一种资源，表示权限和条件要应用的内容。包含子元素描述（可读描述）、元数据（元数据引用）、定位（如何定位作品）和部分内容（部分作品的识别）

续表

类 型	元 素	定 义
	simpleDigitalWork-Metadata	数字作品元数据的集合。包括子元素 title、 creator、 publisher、 publicationDate、 owner 和 copyright
	securityLevel	一个主体具有的安全级别的抽象表示，包括元素值
对 condition 的扩展		
	destination	指定作品可以存放的库。使用 involve 将数字作品移到库中（除了呈现权外的所有权限）
	source	指定操作一个权限时要使用的安全库或设备的源地址。使用除了呈现权之外的所有权限
	helper	指定可以操作一个权限的软件。例如，指定一个特定的播放器来放映一部数字电影
	renderer	指定可以用来呈现一个作品的设备，使用呈现权
	watermark	当生成一个拷贝时，由一个设备嵌入作品拷贝中的信息列表

6.3 核心模式的基本语法

6.3.1 主体

主体是行为方的标识，它包括以下概念类^[6]。

① 一般由资源的提供者（provider）、使用者（customer）、传播者（transmitter）和管理者（manager）对于每种不同的对象封装不同的行为和属性。

② 在 XrML 基于非对称密钥的体系中，把对象定义为密钥的持有者（keyHolder），或多种验证的对象（allPrinciples）。通常这一信息与主体能证明其身份的某些验证机制相关联。

元素 principal 封装了那些被授权的主体标识，指定了用信息唯一标识的交易方。这些信息与某些认证机制有关，以使主体证实其身份。主体类型支持以下的标识技术^[3]。

① 密钥持有者，即拥有密钥（如公钥/私钥对中的私钥）的人。密钥持有者采用 XML 签名中的 keyInfo 来表示。

② 一个主体拥有多个证书，这些证书必须被认证为同时有效的。

③ 其他的标识技术。

XrML 核心模式定义了 Principal（主体）、AllPrincipals（所有主体）、KeyHolder（密钥持有者）几个抽象元素类型。

1. Principal

在 XrML 中，类型 Principal 的实例（或它的派生）表示包含授权或操作权限的实体的唯一标识。

```
Schema Representation of the Principal Type
<xsd:complexType name="Principal">
  <xsd:complexContent>
```

```
<xsd:extension base="r:Resource"/>
</xsd:complexContent>
</xsd:complexType>
```

实际上，Principal 类型是一个概念的抽象，它不指定一个特定的主体是如何被识别或认证的，这个工作由 Principal 的派生类来完成。

2. AllPrincipals 主体

从结构上说，AllPrincipals 类型是一个简单的容器，其中有 0 个或多个 Principal。从语义上说，一个 AllPrincipals 主体 a 表示它所有子节点 Principal 的逻辑连接，即子节点的集合，像整体被识别的实体一样去完成操作。例如，如果 a 在某个 grant 中被识别，当主体要在银行借贷上签名时，从概念上说，要求 a 的每个子节点共同签名。

Schema Representation of the AllPrincipals Type

```
<xsd:complexType name="AllPrincipals">
  <xsd:complexContent>
    <xsd:extension base="r:Principal">
      <xsd:sequence>
        <xsd:elementref="r:principal" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

在这个定义中，AllPrincipals 子节点省缺情况下（即包含 0 个子节点）等价于识别所有可能的主体。AllPrincipals 也可以包含其他的 AllPrincipals。

3. KeyHolder 主体

KeyHolder 类型的实例表示持有某个密钥的实体。例如，使用 KeyHolder 类型，一个采用公钥加密的实体就可以从概念上被认为是“这个实体拥有和这个公钥对应的私钥”（实际上，用这种方法对主体进行标识是很普遍的）。

Schema Representation of the KeyHolder Type

```
<xsd:complexType name="KeyHolder">
  <xsd:complexContent>
    <xsd:extension base="r:Principal">
      <xsd:sequence minOccurs="0">
        <xsd:element name="info" type="dsig:KeyInfoType"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

XrML 2.0 规范本身没有指定 KeyHolder 的加密方法，KeyHolder 的 info 子元素（其类型为 dsig:KeyInfo）由 XrML 2.0 定义，XML 签名语法和处理规范指定了所使用的方法。

6.3.2 权限

XrML 2.0 核心提供了一个<right>元素来封装有关权限的信息。它还提供一套经常用到的特定权限集，如签发、收回、代理与获得权限。对 XrML 核心的扩展可以定义与使用特定类型资源相适合的权限。比如，XrML 内容扩展定义了符合数字作品使用的权限（如播放、打印的权限）。

XrML 核心模式定义了 Right（权限）、Issue（签发）、Revoke（撤销）、PossessProperty（资产拥有）、Obtain（获得）等元素类型。

1. Right

在 XrML 2.0 中，Right 类型（或其派生）的实例表示一个“动词”，指定了一个操作或动作，执行的主体由某个 Grant 授权，主体可以执行这个操作或使用某个相关的目标资源。语义上规定 Right 所指定的资源是其所在的、被授权的 Grant 能合法使用的。

Schema Representation of the Right Type

```
<xsd:complexType name="Right" abstract="false">
  <xsd:complexContent>
    <xsd:extension base="r:LicensePart"/>
  </xsd:complexContent>
</xsd:complexType>
```

实际上，Right 类型是概念上的抽象，也就是说，Right 类型本身并不表示任何实际的操作，这些操作是在 Right 类型的派生类中定义的。这些派生类通常在 XrML 2.0 的扩展中定义，尤其是与特定应用领域相关的权限，但有几个权限是与 XrML 2.0 本身相关的，因此，它们在 XrML 2.0 核心中定义。

2. Issue 权限

Issue 类型的实例为特定资源（通常是授权 Grant）发布许可，体现了许可证书权威方的概念。

Schema Representation of the Issue Type

```
<xsd:complexType name="Issue">
  <xsd:complexContent>
    <xsd:extension base="r:Right"/>
  </xsd:complexContent>
</xsd:complexType>
```

使用 Issue 权限是一个基本的机制，通过这个机制，XrML 2.0 可以由一个许可证授权给另一个许可证。

3. Revoke 权限

Revoke 类型的实例表示撤回先前的声明。例如，发布一个许可的同时也隐含着授权方有撤销权，撤销后授权方可将权限授予其他人。

```
Schema Representation of the Revoke Type
<xsd:complexType name="Revoke">
  <xsd:complexContent>
    <xsd:extension base="r:Right"/>
  </xsd:complexContent>
</xsd:complexType>
```

当使用 Revoke 权限时，XrML 2.0 核心要求被撤销的相关资源要明确标识，但 XrML 2.0 核心本身并不是通过定义明确的 XML 数据类型来完成的，而是选择对核心进行扩展。

4. PossessProperty 权限

PossessProperty 类型的实例表示相关的主体声明拥有某些特性，这些特性常常由一系列资源表示。

```
Schema Representation of the PossessProperty Type
<xsd:complexType name="PossessProperty">
  <xsd:complexContent>
    <xsd:extension base="r:Right"/>
  </xsd:complexContent>
</xsd:complexType>
```

PossessProperty 类型对资源没有限制，但不允许省略。XrML 2.0 核心本身并不定义资源，但是有些资源对 PossessProperty 类型的使用非常有帮助，它们在 XrML 2.0 标准扩展中定义。

5. Obtain 权限

Obtain 权限表示当条件满足后，获得相应的资源。

```
Schema Representation of the Obtain Type
<xsd:complexType name="Obtain">
  <xsd:complexContent>
    <xsd:extension base="r:Right"/>
  </xsd:complexContent>
</xsd:complexType>
```

6.3.3 资源

资源是授权主体执行权限的对象，包括：

- ① 一份数字作品，如电子书、音频或视频文件、图像。
- ② 一项服务，如 Web 网络服务实例或特定的执行一定功能的网络端点（如电子邮件服

务或 B2B 交易服务)。

③ 其他信息。

XrML 2.0 核心为标识与使用特定资源所需信息、与特定模式相匹配的多个资源提供封装。后者允许用一些共同特征来标识一个资源集合。对 XrML 核心的扩展可以定义与特定商业模式相适应的资源。

1. Resource

类型 Resource (或派生) 的实例表示 Grant 中的主体具有某种操作权限的“直接对象”，但是在 XrML 2.0 中，并不是所有操作都需要 Resource 这种直接对象。

Schema Representation of the Resource Type

```
<xsd:complexType name="Resource" abstract="false">
  <xsd:complexContent>
    <xsd:extension base="r:LicensePart"/>
  </xsd:complexContent>
</xsd:complexType>
```

表示资源的 Resource 类型在概念上是抽象的，并不用于任何实际的客体对象，而是用于扩展定义其他具体类型。但有几个资源是与 XrML 2.0 本身相关的，因此，它们在 XrML 2.0 核心中定义。

2. DigitalResource

Grant 中的 DigitalResource 提供了一种方法，将二进制数据流标识为与 Grant 相关的目标对象。尤其重要的是，这些数据不必按照 XML 规范的要求转成字符串，而可以是任意的二进制数据。

Schema Representation of the DigitalResource Type

```
xsd:complexType name="DigitalResource">
  <xsd:complexContent>
    <xsd:extension base="r:Resource">
      <xsd:choice minOccurs="0">
        <xsd:element name="nonSecureIndirect" type="r:NonSecureReference">
        </xsd:element>
        <xsd:element name="secureIndirect" type="dsig:ReferenceType">
        </xsd:element>
        <xsd:element name="binary" type="xsd:base64Binary">
        </xsd:element>
        <xsd:element name="xml">
        <xsd:complexType mixed="true">
          <xsd:sequence>
            <xsd:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs=
              "unbounded"/>
          </xsd:sequence>
        </xsd:complexType>
        </xsd:element>
      </xsd:choice>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```

        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:any namespace="##other" processContents="lax">
</xsd:any>
</xsd:choice>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

6.3.4 条件

条件指定了使权限得以执行的约束。简单的条件是执行权限的时间段；稍微复杂一点的条件，如要求存在一个事先颁发给主体的有效先决的权限，使用这种机制，使得执行某项权限的资格依赖于执行其他权限的资格，而且可以列一个条件清单，要求所有这些条件必须同时满足。

XrML 2.0 核心定义了 Condition（条件）抽象元素类型，以及 AllConditions（所有条件）、ValidityInterval（有效期）、RevocationFreshness（撤销的更新）、ExistsRight（既存权利）、PrerequisiteRight（先决权利）等扩展类型。

1. Condition

在 XrML 2.0 中，Condition 类型（或其派生）的实例表示一个语法上的条件子句，主体必须满足这个条件，才能执行包含这个条件的 Grant。

Schema Representation of the Conditions Type

```

<xsd:complexType name="Condition" abstract="false">
    <xsd:complexContent>
        <xsd:extension base="r:LicensePart"/>
    </xsd:complexContent>
</xsd:complexType>

```

2. AllConditions 条件

从结构上说，AllConditions 条件是一个简单的容器，它包含 0 个或多个条件。从语义上说，AllConditions 条件表示条件的逻辑连接，也就是说，这些条件都必须满足。

Schema Representation of the AllConditions Type

```

<xsd:complexType name="AllConditions">
    <xsd:complexContent>
        <xsd:extension base="r:Condition">
            <xsd:sequence>
                </xsd:elementref="r:condition" minOccurs="0" maxOccurs="unbounded">
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

```

3. ValidityInterval 条件

ValidityInterval 条件表示相邻的、不间断的时间间隔。

Schema Representation of the ValidityInterval Type

```

<xsd:complexType name="ValidityInterval">
  <xsd:complexContent>
    <xsd:extension base="r:Condition">
      <xsd:sequence>
        <xsd:element name="not Before" type="xsd:dateTime" minOccurs="0">
        </xsd:element>
        <xsd:element name="not After" type="xsd:dateTime" minOccurs="0">
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

4. RevocationFreshness 条件

如前所述，XrML2.0 许可证的发布者可能会在许可证中指定他们所用的签名方法，过后签名可能会被撤销。实际的 XrML 2.0 处理系统多采用周期性轮询来传递撤销信息，会有等待迟延，在 Grant 或 GrantGroup 的 RevocationFreshness 条件中可设置等待迟延的上限。

Schema Representation of the RevocationFreshness Type

```

<xsd:complexType name="RevocationFreshness">
  <xsd:complexContent>
    <xsd:extension base="r:Condition">
      <xsd:sequence minOccurs="0">
        <xsd:choice>
          <xsd:element name="maxIntervalSinceLastCheck" type="xsd:duration">
          </xsd:element>
          <xsd:element name="noCheckNecessary">
          </xsd:element>
        </xsd:choice>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>

```

```
</xsd:complexType>
```

如果授权 Grant 或 GrantGroup g 的 RevocationFreshness 条件包含一个 maxIntervalSince LastCheck 元素, 而且指定的持续时间长度 d 大于 0。为了满足 Condition 条件, 实际的时间长度值为从许可证的授权时间开始到最后一次撤销签名的轮询检查时间, 这个值必须小于或等于 d ; 如果持续时间 d 的长度等于 0, 那么必须执行撤销签名的轮询检查。持续时间 d 的长度不能小于 0。

5. ExistsRight 条件

Schema Representation of the ExistsRight Type

```
<xsd:complexType name="ExistsRight">
  <xsd:complexContent>
    <xsd:extension base="r:Condition">
      <xsd:sequence minOccurs="0">
        <xsd:choice>
          <xsd:element ref="r:grant"/>
          <xsd:element ref="r:grantPattern"/>
          <xsd:element ref="r:grantGroup"/>
          <xsd:element ref="r:grantGroupPattern"/>
        </xsd:choice>
        <xsd:element ref="r:trustedIssuer" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6. PrerequisiteRight 条件

PrerequisiteRight 条件与 ExistsRight 条件有关联, 但在很多方面不同。ExistsRight 条件确定某个 Grant 和 GrantGroup 是否直接由某个 trustedIssuer 正确授权; 而 PrerequisiteRight 条件确定 (在某个 trustedIssuer 授权的情况下) 给定的主体对给定的资源具有给定的权限所要满足的条件。

Schema Representation of the PrerequisiteRight Type

```
<xsd:complexType name="PrerequisiteRight">
  <xsd:complexContent>
    <xsd:extension base="r:Condition">
      <xsd:sequence minOccurs="0">
        <xsd:element ref="r:principal" minOccurs="0"/>
        <xsd:element ref="r:right"/>
        <xsd:element ref="r:resource" minOccurs="0"/>
        <xsd:element ref="r:trustedIssuer" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```

        </xsd:sequence>
    </xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

6.3.5 其他内核类型和元素

1. TrustedPrincipal

TrustedPrincipal 类型（或其派生）的元素指定了一个策略，使主体在某种使用环境下被认为是可信的。

Schema Representation of the TrustedPrincipal Type

```

<xsd:complexType name="TrustedPrincipal">
  <xsd:complexContent>
    <xsd:extension base="r:LicensePart">
      <xsd:choice minOccurs="0">
        <xsd:element ref="r:principal"/>
        <xsd:element name="any">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:elementref="r:principal" maxOccurs="unbounded"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:choice>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

在 TrustedPrincipal 中，这种策略通过两种途径指定：

① 如果元素 TrustedPrincipal/principal 存在，那么被识别的主体集合只有这个主体 principal。

② 如果元素 TrustedPrincipal/any 存在，那么被识别的主体集合为里面包含的任何实体。

我们经常会用到这样的表示：TrustedPrincipal 中的主体包含变量引用，这个变量通过 ForAll 元素中的模式指定了一个主体集合。

2. ServiceReference

规范中服务这个术语指的是运行的软件，服务的执行与使用它的客户端软件是不同的。

Schema Representation of the ServiceReference Type

```

<xsd:complexType name="ServiceReference">

```

```

<xsd:complexContent>
  <xsd:extension base="r:Resource">
    <xsd:sequence minOccurs="0">\
      <xsd:choice>
        <xsd:sequence>
          <xsd:element name="wsdl" type="r:DigitalResource">
            </xsd:element>
          <xsd:element name="service" type="xsd:NCName">
            </xsd:element>
          <xsd:element name="portType" type="xsd:NCName" minOccurs="0">
            </xsd:element>
        </xsd:sequence>
        <xsd:sequence>
          <xsd:element name="kind">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="wsdl" type="r:DigitalResource">
                  </xsd:element>
                <xsd:element name="binding" type="xsd:NCName">
                  </xsd:element>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="address">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:any namespace="##other" processContents="lax"/>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
        <xsd:element name="uddi" type="r:UddiServiceIdentifier">
          </xsd:element>
        <xsd:any namespace="##other" processContents="lax">
          </xsd:any>
        </xsd:choice>
      <xsd:element name="serviceParameters" minOccurs="0">
        <xsd:complexType>
          <xsd:sequence minOccurs="0" maxOccurs="unbounded">

```

```

        <xsd:element name="datum">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:any namespace="##any" processContents="lax"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="transforms"
            type="dsig:TransformsType" minOccurs="0">
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

```

ServiceReference实例的作用是指定客户和一个具体的服务进行交互的位置、方法、方式。ServiceReference实例具有如下功能。

- ① 识别服务的位置或地址。
- ② 识别关于服务语义的元数据，以及客户与之交互要遵守的规则。
- ③ 任意指定一个具体参数的集合，当一个客户与服务交互时，通过引用ServiceReference来提供这些参数。这些参数提供了一种方式，能够在运行时区分在不同的 XrML 环境下服务的使用。

XrML 并没有使用一种新的方式去描述服务，相反，它利用了这个领域中已有的方法。下面两种技术，可提供ServiceReference中的位置和元数据。

- ① WSDL，即 Web 服务定义语言。
- ② UDDI，即通用描述、发现与集成服务。

3. LicenseGroup

LicenseGroup 类型的实例是包含许可证的容器。在同一 LicenseGroup 中的两个 License 之间没有内在的语义传递，定义这种类型仅由于使用这种容器很方便。

Schema Representation of the LicenseGroup Type

```

<xsd:complexType name="LicenseGroup">
    <xsd:sequence>
        <xsd:element ref="r:license" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>

```

6.4 XrML 的运行机制

6.4.1 XrML SDK 结构

XrML SDK 的体系是一套全面支持 XrML 2.0 规范，由 XrML 模板、XrML 框架、许可解释器插件和条件验证器插件等组成的体系（图 6-6），他们以 DCOM 的形式分布于 XrML 应用中^[6]。

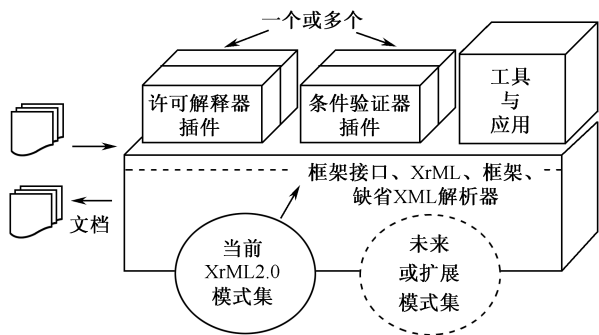


图 6-6 XrML SDK 体系

1. XrML 模板

XrML 模板是一种通用模式，相同或相近的应用场景可以归纳为同一模板。XrML 模板包含一些记号（token），这些记号可以在实际应用时由特定信息替代，从而更加方便地创建完整的 XrML 文件或 XrML 片段（即 XrML 元素）。

2. XrML 框架

XrML 框架处于整个 XrML SDK 的中心位置，将 XML 解析器、许可解释器插件和条件验证器插件等集成到一起。使用这种插件架构，用户可以创建自己的许可解释器插件、条件验证器插件（插件）以及工具与应用。XrML 框架提供了默认 XML 解析器，可以创建、（根据模式）验证、操作和解释 XrML 文档。

XrML 框架提供如下功能：初始化 XrML 应用环境，注册基于条件的事件处理器，对 XrML 文档进行读取、解析、验证、创建、写入等处理，调用许可 XrML 许可解释和 XrML 条件验证。

3. XrML 许可解释

XrML 许可解释是对 XrML 文档的编译生成体系可理解的集合。

4. XrML 条件验证

XrML 条件验证对 XrML 许可解释提供的信息集合在体系内进行测试、验证，它是体系

工作的核心部件。许可解释是条件验证的前提，条件验证是该体系基本的运作核心。

6.4.2 基本流程

XrML 体系的基本流程是对文档的许可解释，以及对解释的条件进行验证的过程。图 6-7 阐述了用户申请执行权利时的基本流程，其中许可（包含一份或多份授权）、权利、主体、资源和条件都使用 XrML 文档表示。

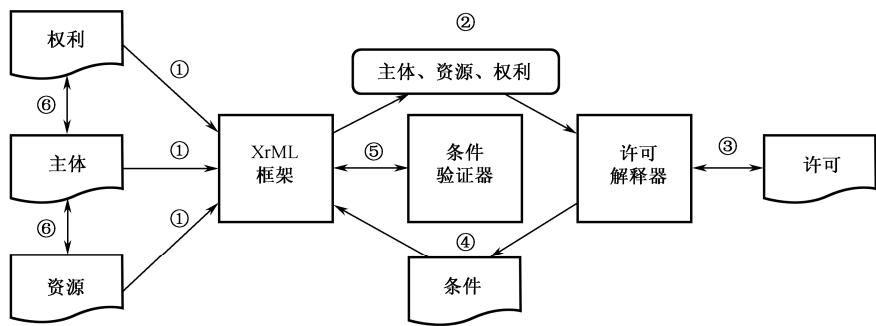


图 6-7 XrML 申请执行权利流程

- ① 特定主体向 XrML 框架请求对特定资源执行特定权限。
 - ② XrML 框架将主体、资源和权限组成查询提交给许可解释器。
 - ③ 许可解释器检查许可，查看许可中是否存在授权与提交的特定主体、资源和权限是否相匹配。
 - ④ 如果许可存在匹配特定主体、资源和权限的授权，则许可解释器返回相应的条件给 XrML 框架。
 - ⑤ XrML 框架调用条件验证器检验返回的许可条件是否满足。
 - ⑥ 如果许可条件满足，XrML 应用允许特定主体对特定资源执行特定权限。
- 以上流程中条件验证是过程的核心。下面进一步分析条件验证的工作流程。

6.4.3 条件验证器行为状态转换机制

XrML 条件验证器包括四个动作，开始（start）、暂停（pause）、恢复（resume）和停止（stop），相应有三种条件验证状态，启动、中断和结束、图 6-8 描述了条件验证器的动作及状态转换图^[6]。

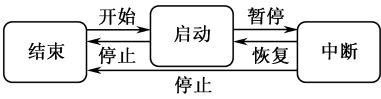


图 6-8 XrML 条件验证器状态转换图

通常，条件验证器使用事件监听器（event listener）向 XrML 框架通知（notify）条件验证结果（条件不满足），而条件的检验和事件的通知在一个周期性执行的定时器（timer）中完成。定时器由开始和恢复动作启动，由暂停和停止动作终止。

6.4.4 条件验证工作流程

条件验证工作流程如图 6-9 所示，它是基本流程的细化，它把许可、解释和验证结合起来了。它在体系中按状态转换机制运作，以保证方案运行的实时性^[6]。

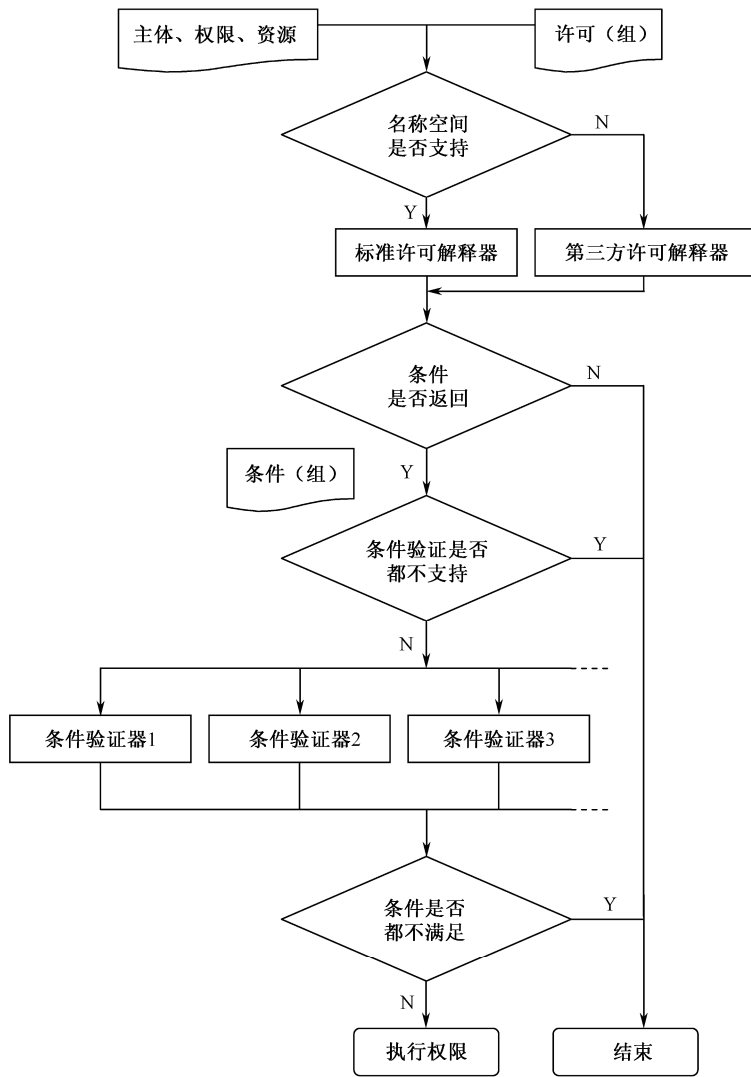


图 6-9 XrML条件验证工作流程图

在条件验证中除了体系固有的插件外，体系允许扩展第三方许可解释器和第三方对非基本名称空间和条件进行解释验证：

- ① 第三方可定义其许可解释器支持第三方特定的名称空间，第三方许可解释器实现为 XrML 框架的插件，所以 XrML SDK 体系中的许可解释器插件有多个。
- ② 在许可（组）中，匹配特定主体、资源和权限的授权可能有多份，相应返回的条件也可能有多个。不同的条件需要定制不同的条件验证器，这些条件验证器实现为 XrML 框架的

插件，所以 XrML SDK 体系中的条件验证器插件有多个。

③ 对于②中返回的多个条件，任何一个条件由所支持的条件验证器检验，如条件满足就可以执行相应权限，而不用多个条件同时满足。

6.5 XML 加密

前面我们提到，权利描述语言 XrML 遵循 XML 加密和数字签名标准^{[7][8]}。本节将对 XML 加密进行介绍。

6.5.1 XML 安全标准概述

XML 安全标准和规范包括：XML 加密（XML Encrypt）、XML 签名（XML Signature）、XML 密钥管理规范（Key Management Specification, XKMS）、安全断言标记语言（Security Assertion Markup Language, SAML）、可扩展访问控制标记语言（Extensible Access Control Markup Language, XACML）、可扩展权利标记语言（Extensible Right Markup Language, XrML）以及 Web 服务安全规范（Web Services Security, WS-Security）等。

W3C 协同 IETF 在 1999 年 6 月开始制定规范，用来支持 XML 语法的加密和数字签名的创建。2002 年 2 月 12 日公布了 XML 数字签名规范。2002 年 9 月公布了 XML 加密规范的推荐标准。

在处理数字签名和加密文档时要用到 PKI，为了方便 PKI 与 XML 应用程序以及使用这些程序的 Web 服务进行集成，2001 年 Microsoft、Verisign 和 WebMethods 共同开发了开放的 XKMS 规范。随后，该规范被提交到 W3C，W3C 组建了一个 XML 密钥工作组（XML Key Management Working Group），以协同其他感兴趣的参与者对其做进一步的开发，2003 年 4 月 18 日发布 2.0 版本。XKMS 与 XML 签名、XML 加密结合，对密钥、证书进行管理，包括注册、分发、撤销等，它还允许客户通过 Web 服务取得密钥信息。

SAML 是交换验证和授权信息的 XML 框架，用于在不同的安全域（security domain）之间交换认证和授权数据。SAML 标准定义了身份提供者（identity provider）和服务提供者（service provider），这两者构成了不同的安全域。它定义了对验证、属性和授权信息进行 XML 编码的语法和语义以及这些安全信息的传输协议。SAML 是结构化信息标准促进组织（Organization for the Advancement of Structured Information Standards, OASIS）安全服务技术委员会（Security Services Technical Committee）制定发布的标准，2003 年 9 月发布了 1.1 版本。

XACML 是 OASIS 访问控制标记语言技术委员会（Access Control Markup Language Technical Committee）制定的 XML 规范，是一种可扩展的访问控制策略语言，用于以 XML 表示访问对象的授权策略。2003 年 2 月 18 日发布了 1.0 版本。

为了加强 Web 服务的安全性，IBM、Microsoft、Verisign 公司于 2002 年 4 月 5 日联合发布了 WS-Security 规范 1.0 版本，提供了一个 SOAP 扩展的标准集合，这个集合可以用于构造安全的 Web 服务时对一致性和机密性的实现。

表 6-4 显示了 XML 相关安全标准。

表 6-4 XML 相关安全标准

安全规范	发展组织	说明
XML Encryption	W3C XML Encryption Working Group	XML 加密规范，实现 XML 数据加密及解密
XML Signature	IETF/W3C XML Signature Working Group	XML 数字签名规范，实现身份验证、保证数据完整性和不可否认性等
XKMS 2.0	W3C XKMS Working Group (WG)	XML 分发和注册公钥信息的规范，适合与已被提议的标准 XML Signature 和 XML Encryption 联合使用
SAML 2.0	OAIS Security Services TC(Technical Committee)	提供标准的方法在 XML 文档中定义用户的认证信息、授权信息等，在可信任域之间交互认证和授权信息
XACML 2.0	OASIS eXtensible Access Control Markup Language TC	XML 访问控制标记语言，基于 XML 规范来表达在网上进行信息访问的策略
XrML	ContentCuard eXtensible Right Markup Language TC	XrML 用于指定使用某资源时相应的权限和条件，其语法用 XML Schema 描述
WS-Security 1.1	OASIS Web Service Security (WSS) TC	以 XML 为基础的 Web 服务安全规范

6.5.2 XML 加密和传统加密的区别

1. 加密粒度不同

XML 加密与传统加密的最大区别是引入了加密粒度的概念，XML 文档作为一种结构化数据，能对加密的粒度进行控制，即加密粒度可选。XML 文件可以：① 加密整个文件；② 加密文件中的元素；③ 加密文件中元素的内容；④ 超级加密，即对加密过的元素或内容进行再加密。这样我们就能选择对文件内的不同信息进行不同的加密处理，以控制对不同元素的授权查看。

例如，对于某个客户的购买信息，商场只需要知道客户的名称和地址，无须知道信用卡的信息，而银行无须知道购买货物的详细信息；对住院看病的患者来说，医院的研究人员无须知道个人住院治疗的详细信息，但管理人员可能正需要这些信息，而医生或护士可能还需要知道部分个人信息。

此外，XML 加密还能处理非 XML 数据，如二进制数据、HTML 文件、JPG 文件等。

2. 加密状态持久、保证数据安全

XML 文档一经加密，在解密之前，不论是存储于磁盘空间中，还是在网络的传输过程中，或是在某个网络节点停留时，都处于加密状态，未经授权无法访问到密码信息，能确保数据安全性^[10]。

3. 可实现多方安全会话

使用 XML 加密，每一方都可以保持与任何通信方的安全或非安全状态，可以在同一文档中交换安全的和非安全的数据。XML 加密可实现多方安全通话。

例如，一个包含许多聊天室的安全聊天应用程序，其中每个聊天室都有几个人。可以在

聊天伙伴之间交换 XML 加密的文件，这样一个聊天室中的数据对其他聊天室而言是不可见的。

4. 满足各种应用环境的需求

XML 加密是以 XML 形式表现被加密的数据的，加密过程不会改变文档的格式，加密后的 XML 文档仍是一个格式良好的 XML 文档。由于 XML 文档在加密的过程中数据格式一致，可以方便地被基于 XML 的应用系统直接处理，无需格式转换。

XML 能满足各种应用环境的共性需求，XML 既可以应用于消息传输，也可以应用于文档数据的存储，并支持特定应用的特殊需求，具有扩展能力。

6.5.3 XML 加密规范和基本结构

在 W3C 制定的 XML 加密规范是加密 XML 数据、以标准 XML 格式表示加密结果以及解密器处理过程的一套标准方法。XML 加密支持目前流行的一系列加密算法，如对称加密（包括分组加密、流加密）、不对称加密、消息摘要等。

在进行 XML 加密时，采用标准的 XML 标记语法来表示相关信息、算法以及实际加密的数据，并以标准的 XML 格式来表示加密结果，既可以是直接含有加密数据的 XML 文档，也可以通过外部引用的加密数据，而且应用程序能方便地访问和处理被加密的数据。

1. XML 加密的名称空间

XML 加密规范定义了如下所示的名称空间：

```
xmlns:xenc='http://www.w3.org/2001/04/xmenc#'
```

XML 名称空间除了提供语法的作用域外，还被用做规范中引用的算法标识符的前缀，能很快识别出被标识的算法（如 RSA、3DES、SHA-1 等）。下面列出一些在 XML 加密规范中定义的算法标识符示例：

```
http://www.w3c.org/2001/04/xmenc#rsa-1_5
http://www.w3c.org/2001/04/xmenc#tripleDES-cbc
http://www.w3c.org/2001/04/xmenc#sha1
```

2. XML 加密元素

XML 加密生成的文件是格式正规的 XML 文件，XML 加密元素包括<EncryptedData>元素及其子元素。<EncryptedData>元素的结构如下（这里 "?" 表示出现 0 次或 1 次； "+" 表示出现 1 次或多次； "*" 表示出现 0 次或多次； "|" 表示选择；空元素标记意味着元素必须是空的）^[7]：

```
<EncryptedData Id? Type? MimeType? Encoding?>
  <EncryptionMethod/>?
  <ds:KeyInfo>
    <EncryptedKey>?
    <AgreementMethod>?
```

```

        <ds:KeyName>?
        <ds:RetrievalMethod>?
        <ds:*>?
    </ds:KeyInfo>?
    <CipherData>
        <CipherValue>?
        <CipherReference URI?>?
    </CipherData>
    <EncryptionProperties>?
</EncryptedData>

```

(1) <EncryptedData>元素

<EncryptedData>元素是封装加密或解密所需相关信息的最外层元素。如果该元素是 XML 文档的根元素，则整个文档都被加密。

使用<EncryptedData>元素构建加密数据，该元素包含与加密和解密信息相关的数据，如加密密钥的信息（使用<ds:KeyInfo>元素）、算法信息（使用<EncryptionMethod>元素）、加密数据（使用<CipherData>元素）以及加密数据的引用（使用<CipherReference>元素）等。在生成的加密数据文件中，<EncryptedData>元素用来替代加密的数据，若加密数据是某个 XML 文件本身，<EncryptedData>元素则成为文件的根元素；若加密数据是 XML 文件的内部元素，则该元素和它的内容一起被删除并用<EncryptedData>元素替代。

XML 加密也能对其他非 XML 文件进行加密，如 HTML 文件、JPG 文件等，其加密方式与加密整个 XML 文件一样，但要先转换为 Base64 格式。

(2) <EncryptionMethod>元素

<EncryptionMethod>元素是<EncryptedData>元素的子元素。如果没有提供这个元素，那么参与 XML 加密的应用程序一定以某种方法共享或者隐含地知道所使用的加密算法。

(3) <ds:KeyInfo>元素

<ds:KeyInfo>元素是<EncryptedData>元素的子元素，该元素提供用于加密和解密数据的对称会话密钥。如果没有提供这个元素，那么参与 XML 加密的应用程序一定以某种方法共享或者隐含地知道所使用的加密算法。

(4) <EncryptedKey>元素

<EncryptedKey>元素是<ds:KeyInfo>元素的子元素，该元素用于交换对称会话密钥。

(5) <AgreementMethod>元素

<AgreementMethod>元素是<ds:KeyInfo>元素的子元素，该元素用于建立一个应用程序定义的、共享会话密钥的方法。如果没有提供这个元素，那么参与 XML 加密的应用程序必须以某种方式来处理密钥协议。

(6) <ds:KeyName>元素

<ds:KeyName>元素是<ds:KeyInfo>元素的子元素，可以用该元素选择使用易读的名字来访问会话密钥。

(7) <ds:RetrievalMethod>元素

<ds:RetrievalMethod>元素是<ds:KeyInfo>元素的子元素,该元素能够提供到另一个含有私有会话密钥的<EncryptedKey>元素的 URI 链接。

(8) <CipherData>元素

<CipherData>元素是<EncryptedData>元素必需的子元素,该元素包含或引用实际的加密数据。如果这个元素包含加密的数据,就会使用<CipherValue>子元素;如果这个元素引用加密的数据,就会使用<CipherReference>子元素。

<CipherData>元素是<EncryptedData>元素的唯一不可选子元素,该元素是有意义的,这是因为<EncryptedData>元素一定会提供加密的数据。

(9) <CipherValue>元素

<CipherValue>元素封装了实际的加密数据。

(10) <CipherReference>元素

<CipherReference>元素封装了对外部加密数据的引用。

(11) <EncryptionProperties>元素

<EncryptionProperties>元素提供了应用程序专用的附加信息,如加密操作的起源、日期和时间,这些信息可能非常有用。

6.5.4 XML 加密粒度的选择

如前所述,XML 加密根据需要可以有不同的粒度,本节将以下面的 XML 文档为例,对 XML 的加密方法进行介绍^[12]。

```
<purchaseOrder>
  <Order>
    <Item>book</Item>
    <Id>123-958-74598</Id>
    <Quantity>12</Quantity>
  </Order>
  <Payment>
    <CardId>123654-8988889-9996874</CardId>
    <CardName>visa</CardName>
    <ValidDate>12-10-2004</ValidDate>
  </Payment>
</purchaseOrder>
```

在上面的 XML 文档中 <Order>元素包含交易商品的具体信息,像商品名 <Item>、商品编号<Id>以及商品数量<Quantity>。<Payment>元素包含付款相关的信息,如信用卡号<CardId>、信用卡公司名称 <CardName>以及信用卡的有效期 <ValidDate>。绝大部分的应用都会认为 <Payment> 元素是敏感信息,但是对于 <Order> 元素的看法可能会不一样,有些公司可能会认为是敏感信息,而有些则不这么认为。因此在处理加密与签名时所针对的元素

也会不一样。在本例中，将会对 <Order> 元素使用不同的加密方案，在针对元素的加密中，可以选择对 <Order> 元素加密和签名，也可以选择对 <Order> 元素保持明文。在针对整个文档的加密方案中，将会对 <Order> 元素和 <Payment> 全部进行加密与签名。

1. 整个 XML 文档的加密与解密

整个 XML 文档的加密，首先将整个 XML 文档转换成字节流，然后对字节流进行加密，这里采用三重 DES (Triple-DES) 的加密块链接 (CBC) 模式算法进行加解密。然后将加好密的字节流再进行 Base64 编码，用 Base64 对加密后的字节流编码是因为在基于 ASCII 字符的 XML 文档中是不能直接载有二进制数据的。经过 Base64 编码后得到了加好密的可以在 XML 文档中进行传输的字符流。这时的字符流不能直接进行传输，因为传输的节点不能根据加密后的数据解释出传输目标在哪里，也不知道怎样对文档进行解析，简单地说，就是加密后的字符流已经不再是一个可以被识别的 XML 文档了。因此我们要将加密后的 XML 文档进行封装，封装好的 XML 文档应该与下面的文档类似。

```
<EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
  Type='http://www.isi.edu/in-notes/iana/assignments/media-types/text/xml'>
  <CipherData>
    <CipherValue>A23B45C56.....</CipherValue>
  </CipherData>
</EncryptedData>
```

在上面的文档中元素<EncryptedData>包含一个名称空间的属性 xmlns，包含一个类型属性 Type，它能告诉解析器该元素里面的数据在加密前是什么类型。它的第一级子节点是<CipherData>元素，从字面意思可以知道它应该是保存加密后的数据的地方。它也包含一个子节点<CipherData>元素，该子节点的包含的内容就是原来的 XML 文档加密后的数据。

本例中，假定加密密钥已经通过密钥交换成功得到了。在上述文档中，<EncryptedData>元素的名称空间为固定的值，是不能改变的；原数据类型可以被定义在 Type 中，为 TEXT/XML。该类型的值在 IANA (Internet Assigned Numbers Authority, 因特网编号管理局) 中定义。在得到<EncryptedData>元素之后，要想在 SOAP 消息中使用，还必须将其加入 SOAP 的消息体中，使之成为 SOAP 的体元素之一。封装好的消息将会成显示成如下 XML 文件。

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <xenc:EncryptedData
      xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
      Id="encrypted-body-entry"
      Type="http://www.w3.org/2001/04/xmlenc#Element">
      <xenc:CipherData>
        <xenc:CipherValue>A23B45C56...</xenc:CipherValue>
      </xenc:CipherData>
```



```

</xenc:EncryptedData>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

从上述文档中可以看出，加密数据前多了一个前缀 **xenc**，它告诉解析器所有带了该前缀的节点都是与加密相关的信息。除了该前缀，其他信息与上面介绍的完整加密的 XML 文档完全一致。从中我们还可以看出所有的加密相关的信息都存放在 SOAP 的消息体元素中，并包含在 SOAP 信封中。

当服务使用者拿到 SOAP 消息后必须先对消息进行解密才能正常使用 XML 文档。首先从 SOAP 消息中获得<EncryptedData>元素，并从中取出<CipherValue>元素的内容。然后根据事先交换好的密钥对<CipherValue>元素的内容进行解密。首先将该内容用 Base64 进行解码，得到字节流后再进行三重 DES 解密，这时我们便可以得到完整的 XML 文档的明文。

2. XML 文档元素的加密与解密

对 XML 文档元素的加密与解密过程与整个文档的加解密过程相似。只是在转换字节流前要先将整个元素转换成字符串，然后再对该字符串进行加密、Base64 编码等操作，最后拿到该元素的密文。如果要对该密文进行传输，同样我们需要将该密文加入 SOAP 的消息体中，使之成为 SOAP 的体元素之一。封装好的 SOAP 消息在结构上与将完整的 XML 文档密文封装好后的 SOAP 消息是一样的。解密过程也类似，这里不再赘述。加密好的 XML 文档应该如下面文档所示。

```

<PurchaseOrder>
  <Order>
    <Item>book</Item>
    <Id>123-958-74598</Id>
    <Quantity>12</Quantity>
  </Order>
  <EncryptedData Type='http://www.w3.org/2001/04/xmenc#Element'
    xmlns='http://www.w3.org/2001/04/xmenc#'>
    <CipherData>
      <CipherValue>A23B45C564587...</CipherValue>
    </CipherData>
  </EncryptedData>
</PurchaseOrder>

```

将上述 XML 文档与整体加密的 XML 文档比较会发现，由于只是选择了<Payment>元素进行加密，上述文档保留了绝大部分的原文档信息，只是原来<Payment>元素被换成了<EncryptedData>元素，结构也与整体加密的 XML 文档类似，这里不再赘述。

如果要在 SOAP 消息中使用该加密后的 XML 文档，同样需要把该元素加入 SOAP 中作为一个体元素。使用的时候将该体元素取出来并进行解密。

3. XML 文档元素内容的加密与解密

元素内容的加解密与 XML 文档元素的加解密相似，在 SOAP 消息中的使用也与 XML 文档元素的使用相似，这里只给出对元素内容加密后的 XML 文档。

```
<PurchaseOrder>
  <Order>
    <Item>book</Item>
    <Id>123-958-74598</Id>
    <Quantity>12</Quantity>
  </Order>
  <Payment>
    <CardId>
      <EncryptedData
        Type='http://www.w3.org/2001/04/xmlenc#Content'
        xmlns='http://www.w3.org/2001/04/xmlenc#'>
        <CipherData>
          <CipherValue>A23B45C564587...</CipherValue>
        </CipherData>
      </EncryptedData>
    </CardId>
    <CardName>visa</CardName>
    <ValidDate>12-10-2004</CardName>
  </Payment>
</PurchaseOrder>
```

从上面的文档可以看出，文档需要加密的部分被替换成了 EncryptedData 元素，当解析器遇到该元素，便知道这里面包含有加密信息，需要解密才能进行正常使用。

4. 超级加密

XML 加密除了能对文件的部分内容进行加密之外，还能通过超级加密，对加密后的数据进行再加密，这种加密方式能够控制文件选定部分内容的查阅权限，将数据资料传递给不同的当事人并能保证数据的保密性。例如，当传递一份订单资料给某公司，需经过公司的销售部门及财务部门，可以先利用财务部门的密钥去针对付款的元素部份加密，形成一个包含元素加密的 XML 文件，然后再利用销售部门的密钥将这份文件整体加密，形成一个超级加密的 XML 文件。当订单传递给销售部门时，销售部门对加密文件进行解密，也不会看到付款部分的信息，直到该文件被传到财务部门后。才能了解整份文件的内容。

一份 XML 文件可以包含多个 <EncryptedData>元素，但是<EncryptedData>不能成为另一个<EncryptedData>的父元素或子元素，因此超级加密必须要加密 <EncryptedData>元素包含的所有内容，不能只加密 <EncryptedData>元素的子元素或元素的内容。

从上面描述可以看出一个共同的特征，所有需要加密的数据在加密后都会被 <EncryptedData>元素替换掉，而其他的元素保持不变。

6.6 XML 数字签名

6.6.1 XML 签名概述

XML 签名规范是对现有数字签名技术的扩展，定义了一套用 XML 表示的数字签名的方法。W3C 将 XML 数字签名解释为：定义一种与 XML 语法兼容的数字签名语法描述规范，描述数字签名本身和签名的生成与验证过程。作为一个安全有效的数字签名方案，该规范提供了数字签名的完整性(Integrity)、签名确认(Authentication)和不可抵赖性(Nonerepudiation)。

为了实现用 XML 来表示数字签名，在 XML 规范中定义了〈Signature〉元素。该元素包括了签名使用的类型、对已签名数据的引用以及验证签名所需的密钥的详细信息。XML 提供了灵活的数字签名机制，在 XML 文档中可以签名任何数据，包括一个完整的 XML 文档、一个 XML 文档的元素或者一个 XML 元素的内容。

XML 数字签名的名称空间为：

```
xmlns:dsig='http://www.w3.org/2000/09/XMLdsig#'
```

XML 签名的产生开始于将要被签名的数据对象的摘要的生成。所有的关于定位原数据对象和关联摘要的信息都存在于 Reference 元素中。根据数据对象与〈Signature〉元素的关联方式，XML 签名分为三种不同的类型，分别为：

① 封外签名 (Enveloping Signature, 也被称为封装式签名)

数据对象与〈Signature〉元素放在同一个文档中，是〈Signature〉元素的一部分，即被签名对象是〈signature〉元素的子元素。

② 封内签名(Enveloped Signature, 也被称为被封装式签名)

数据对象与〈Signature〉元素放在同一个文档中，〈Signature〉元素是被签名对象的子元素。

③ 分离签名 (Dctached Signature)

数据对象与〈Signature〉元素各自独立存在，可以属于同一个文档，也可以在另一个完全不同的文档中。

6.6.2 XML 签名的基本结构和语法

XML 数字签名由〈Signature〉元素表示，其结构如下(这里 "?" 表示出现 0 次或 1 次；"+" 表示出现 1 次或多次； "*" 表示出现 0 次或多次)^[8]：

```
<Signature ID?>
  <SignedInfo>
    <CanonicalizationMethod/>
    <SignatureMethod/>
    (<Reference URI? >
```

```

        (<Transforms>)?
        <DigestMethod>
        <DigestValue>
    </Reference>)+
</SignedInfo>
<SignatureValue>
(<KeyInfo>)?
(<Object ID?>)*
</Signature>

```

(1) <Signature>元素

<Signature>元素是 XML 签名的根元素。

(2) <SignedInfo>元素

<SignedInfo>元素的子元素包含签名数据以及签名验证的其他附加信息。签名算法实际上应用于该元素及其所有子元素以生成签名。

(3) <CanonicalizationMethod>元素

<CanonicalizationMethod>元素指定了产生<SignedInfo>元素的规范形式的算法。

(4) <SignatureMethod>元素

<SignatureMethod>元素指定了<SignedInfo>元素的签名算法。

(5) <Reference>元素

<Reference>元素包含 URI 属性，用于识别被签名的数据对象。

(6) <Transforms>元素

每个<Reference>元素都可有 0 个或多个指定的转换，按照它们在列出的顺序，对<Reference>元素中 URI 属性指定的数据对象进行一系列转换。

(7) <DigestMethod>元素

<DigestMethod>元素指定产生数据对象摘要信息的算法。

(8) <DigestValue>元素

<DigestValue>元素包括数据对象的摘要信息。

(9) <SignatureValue>元素

<SignatureValue>元素存储了<SignedInfo>元素及其子元素计算得到的数字签名值。

(10) <KeyInfo>元素

<KeyInfo>元素存储了接收者得到数字签名验证密钥的附加信息。

(11) <Object>元素

<Object>元素为可选元素，存储了封装签名或数据对象。

下面是一个 XML 数字签名的例子^[8]：

```

<Signature Id="MyFirstSignature" xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>

```

```

<CanonicalizationMethodAlgorithm="http://www.w3.org/2006/12/xml-c14n11"/>
<SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
<Reference URI="http://www.w3.org/TR/2000/REC-xhtml1-20000126/">
  <Transforms>
    <Transform Algorithm="http://www.w3.org/2006/12/xml-c14n11"/>
  </Transforms>
  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
  <DigestValue>dGhpcyBpcyBub3QgYSBzaWduYXR1cmUK...</DigestValue>
</Reference>
</SignedInfo>
<SignatureValue>...</SignatureValue>
<KeyInfo>
  <KeyValue>
    <DSAKeyValue>
      <P>...</P><Q>...</Q><G>...</G><Y>...</Y>
    </DSAKeyValue>
  </KeyValue>
</KeyInfo>
</Signature>

```

6.6.3 创建 XML 签名

下面是创建 XML 数字签名的过程：

(1) 生成<Reference>元素

对要签名的数据对象：

- ① 按照<Transform> 元素出现的顺序和指定的算法对数据对象进行转换。
- ② 用<DigestMethod>元素指定的算法对转换好的数据对象进行 Hash 运算，并将转换好的 Hash 值存储在<DigestValue>素中。
- ③ 创建<Reference>元素，该元素包括可选的数据对象标识符、可选的<Transform>元素、<digestMethod>元素和<Digestvalue>元素。

(2) 生成签名

- ① 创建<SignedInfo>元素，该元素包括<SignatureMethod>、<CanonicalizationMethod>、和<Reference>子元素。
- ② 按照<CanonicalizationMethod>元素中指定的算法对<SignedInfo>元素进行规范化，并使用<SignatureMethod>元素中指定的算法计算签名值，签名值存储在<SignatureValue>元素中。
- ③ 创建<Signature>元素，包括<SignedInfo>、<Signature>和可选的<Object>、<KeyInfo>、等子元素。

6.6.4 验证 XML 签名

当接收方收到 XML 签名，就可以进行验证。XML 数字签名的验证过程分为两部分：引用确认（ReferenceValidation，也称参考确认或内容确认）和签名确认（Signature Validation）。引用确认的目的是确保被签名对象没有被任何的修改，而签名确认的目的是保证签名者身份的真实性。

XML 数字签名的验证过程如下：

（1）引用确认

根据<CanonicalizationMethod>对<SignedInfo>做规范化处理，并对于在<SignedInfo>中的每一个<Reference>元素进行如下操作，这一步骤即引用确认。

- ① 通过 URI 等方式从中获得需要被计算摘要的数据，并对其应用<Transforms>转换规则。
- ② 根据<DigestMethod>的算法计算出摘要值。

③ 将生成的摘要值与<DigestValue>的值进行比较，如果不同，则认证失败，说明被签署的对象已经被修改过；否则进行签名确认。

（2）签名确认

- ① 从<KeyInfo>或者外部源得到密钥（Key）的相关信息。
- ② 用<CanonicalizationMethod>中的算法对<SignedInfo>元素进行规范。

③ 通过<SignatureMethod>，使用密钥信息和已在第②步规范化的<SignedInfo>元素进行计算，并将计算结果与<Signaturevalue>比较，如果相同，则校验成功，数字签名有效；否则签名认证失败。

6.7 ODRL

6.7.1 ODRL 模型

1. ODRL 基础模型

ODRL 是 DRM 领域描述数字内容权限的语言。它旨在提供灵活的、可互操作的机制，它以 XML 作为语法，包括一组核心实体以及它们之间的关系。ODRL 基础模型如图 6-10 所示^{[5][9]}。

ODRL 基础模型由以下三个核心实体组成。

资源（Asset）：包括实物和数字化内容。一个资源必须被唯一认证，它可能由很多子部分组成，也可能有其他不同形式。资源也可能被加密以保证内容安全发布。

权限（Rights）：包括许可（Permission），许可又包括约束（Constraint）、需求（Requirement）和条件（Condition）。许可是在资源上被允许的操作或行为。约束是对许可的限制（如最多播放 5 次视频），需求是为了执行许可而必须完成的义务（如每次播放视频须付 5 元），条件则

例外，即条件为真，则许可过期，需要重新协商（比如信用卡过期，则所有的许可都不能播放视频）。

主体（Party）：包括终端用户和权限持有者（Rights Holders）。主体可以是人、组织或者定义的角色。终端用户通常是资源的消费者；权限持有者通常是在资源的创建、生产和分发中扮演重要角色的团体，对资源具有某种形式的所有权。

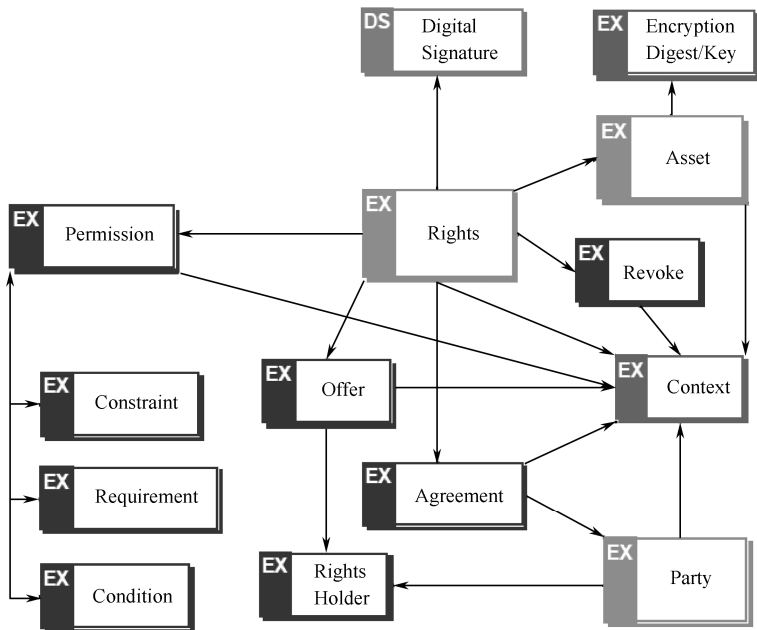


图 6-10 ODRL基础模型

利用这三个实体，基础模型可以表示提议（Offer）和协议（Agreement）。提议是权限持有者对他们的资源所拥有的特定权利的建议。协议是主体签订合同或处理特定提议时达成的一致性约定。模型也可以表示对任何提议和协议的撤销。

ODRL 基础模型可以用 XML 来表示，下面这个例子是用 XML 语法表示的基础模型。

```
<rights>
  <context>
    <uid>...</uid>
  </context>
  <offer>
    <asset>...</asset>
    <permission>
      <permission-type>
        <requirement>...</requirement>
        <constraint>...</constraint>
      </permission-type>
      <condition>...</condition>
    </permission>
  </offer>
</rights>
```

```

    <party>
      <context>...</context>
      <rightsholder>...</rightsholder>
    </offer>
  <agreement>
    <context>...</context>
    <party>...</party>
    <permission>...</permission>
    <asset>...</asset>
  </agreement>
</rights>

```

<rights>是 ODRL 的根元素，<rights>包括<offer>和<agreement>等子元素。<party>用于说明权利所有人，数据对象为<asset>。ODRL 包含权利所有人的权利要求标签<offer>和权利双方遵守的权利协议<agreement>。在 ODRL 中，<permission>是数据对象<asset>执行某些操作的权限，如 play、print、display 和 execute 等。<permission>由限制标签<constraint>和需求标签<requirement>说明，限制标签<constraint>说明用户在执行一定操作前必须满足的条件，需求标签<requirement>说明用户在执行一定操作前必须采取的额外操作（如付费等）。

下面是一个 ODRL 的例子。

```

<rights>
  <agreement>
    <asset>
      <context>
        <uid>example</uid>
      </context>
    </asset>
    <permission>
      <play>
        <constraint>
          <hardware>
            <context>
              <uid>pl</uid>
            </context>
          </hardware>
        </constraint>
      </play>
    </permission>
  </agreement>
</rights>

```


2. ODRL 许可模型

ODRL 支持表示许可的提议（Offer）和协议（Agreement），这是在资源上所认可的操作的集合。ODRL 的许可模型如图 6-11 所示^[5]。

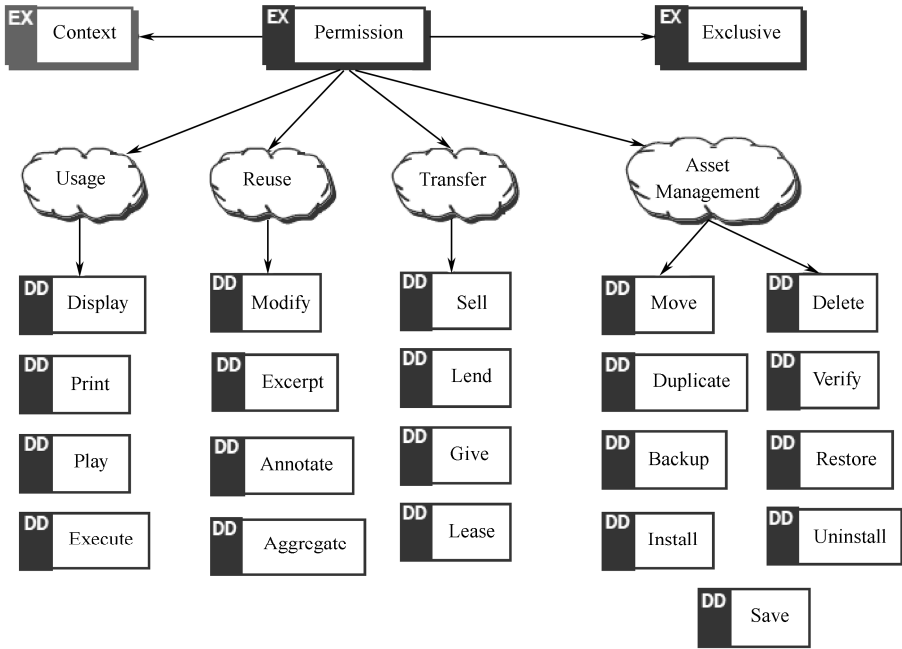


图 6-11 ODRL 许可模型

许可 Permission 实体包括四个抽象实体，和许可 Permission 一样，抽象实体在组中单独使用，抽象实体包括如下内容。

- ① Usage: 定义了一个方法的集合，使用这些方法可以消费资源（如显示 Display、打印 Print、播放 Play、执行 Execute）。
- ② Reuse: 定义了一个操作的集合，使用这些操作（或其中一部分）资源可以重新使用（如修改 Modify、摘录 Excerpt、注释 Annotate、统计 Aggregate）。
- ③ Transfer: 定义了一个过程的集合，这些过程资源上的权利可以转移（如卖 Sell、出借 Lend、给予 Give、租借 Lease）。
- ④ Asset Management: 定义了一个数字资源管理操作的集合（如移动 Move、复制 Duplicate、删除 Delete、验证 Verify、备份 Backup、恢复 Restore、存储 Save、安装 Install、卸载 Uninstall）。

另外，许可 Permission 支持如下内容。

- ① Exclusive 属性：授权许可是否限制在指定的主体。
- ② Context 实体：用于赋予特定的集合或权限组唯一的标识。

许可 Permission 必须通过一个 Offer 或 Agreement 与一个或多个 Asset 发生联系，这种联系可以被扩展（即许可 Permission 是 Offer 或 Agreement 的孩子）或限定（通过一个 Offer 或 Agreement 的引用）。一个许可可以与零个或多个约束 Constraint、条件 Condition 和需求 Requirement 相联系。

下面是一个许可的例子。

```
<permission>
  <display/>
  <print>
    <constraint>...</constraint>
  </print>
  <annotatel/>
</permission>
```

3. ODRL 约束模型

约束 Constraint 用于限制相关权限的使用，ODRL 支持权利约束表示，这是一个被认可的在资源许可上的限定集，ODRL 约束模型如图 6-12 所示^[5]。

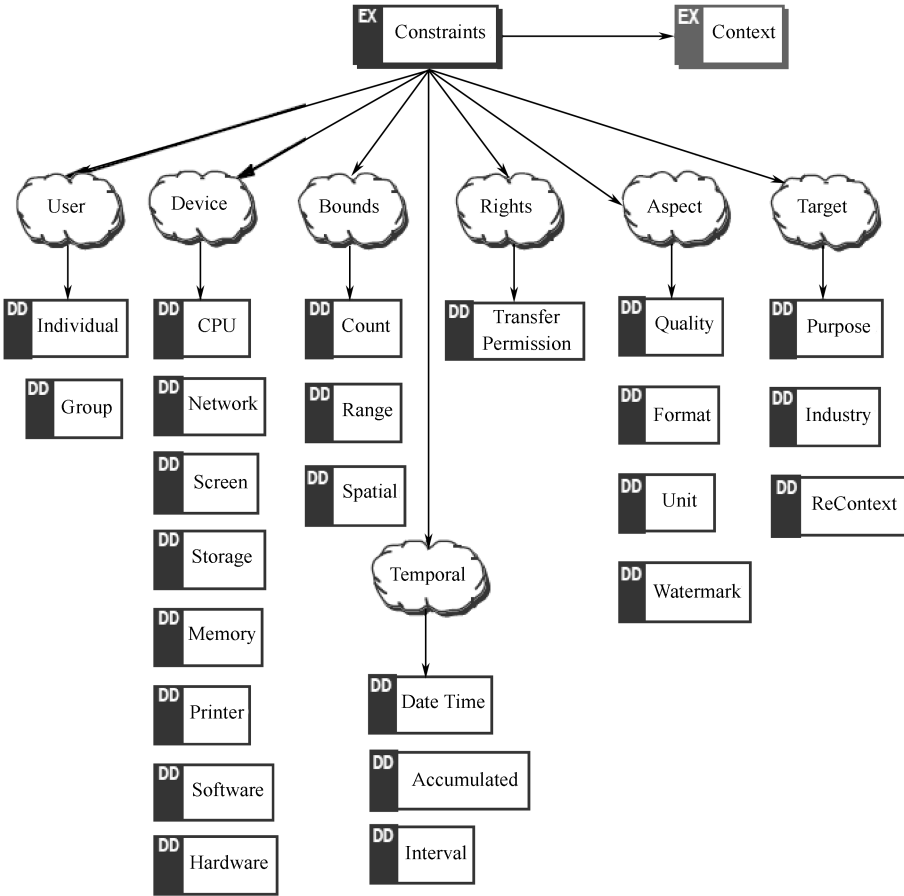


图 6-12 ODRL约束模型

Constraint 实体由几个抽象实体聚合而成，包括如下内容。

- ① User: 定义了一个约束的集合来限定可标识用户的使用（如个体、组）。
- ② Device: 定义了一个约束的集合来限定物理设备或系统的使用（如 CPU、网络、屏幕、

外存、内存、打印机、软件、硬件)。

③ **Bounds**: 定义了一个约束的集合来限定对一个固定数字或长度/面积的使用 (如数量、范围、空间)。

④ **Temporal**: 定义了一个约束的集合来限定对时间范围的使用 (如日期时间、累积、间隔)。

⑤ **Aspect**: 定义了一个约束的集合来限定对资源的不同特征或表示的使用 (如数量、格式、单位、水印)。

⑥ **Target**: 定义了一个约束的集合来限定资源在哪里如何使用。

⑦ **Rights**: 定义了一个约束的集合, 只用于具有转移许可的资源, 并且使得说明 (以及约束) 应用在下载的许可上 (如转移许可)。

下面是 ODRL 约束模型的例子。在这个例子中, 显示 **display** 许可约束到一个特定的 CPU 上, 打印 **print** 许可约束只能打印 5 次, 播放 **play** 许可限制在 7 天, 并且只能播放 10 次。

```
<display>
  <constraint>
    <cpu/>
  </constraint>
</display>
<print>
  <constraint>
    <count>5</count>
  </constraint>
</print>
<play>
  <constraint>
    <interval>P7D</interval>
  </constraint>
  <constraint>
    <count>10</count>
  </constraint>
</play>
```

4. ODRL 需求模型

ODRL 支持权利需求的表示, 这是一个在获取相关许可前必须满足的预条件集。ODRL 需求模型如图 6-13 所示^[5]。

需求 Requirement 实体包括三个抽象实体。

① **Fee**: 定义了一个付费使用的需求集合 (如 PrePay、PostPay、PerUse)。

② **Interactions**: 定义了一个用户交互的需求集合 (如 Accept、Register)。

③ **Usage**: 定义了一个资源使用的需求集合 (如 Attribution、Tracked)。

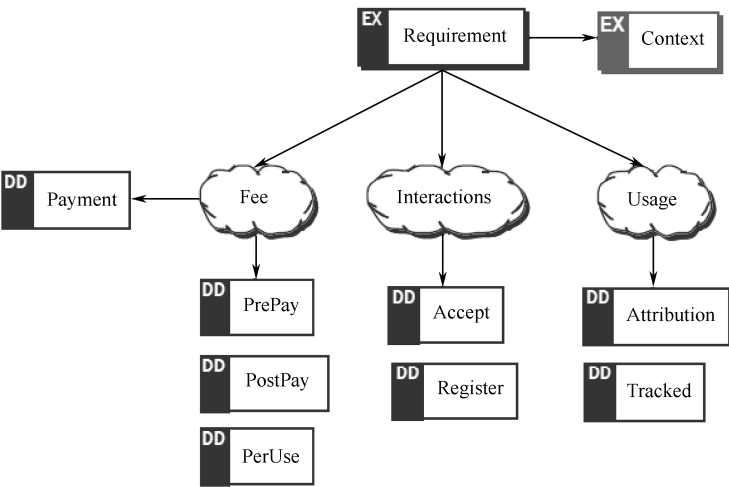


图 6-13 ODRL需求模型

下面是一个 ODRL 需求模型的例子。在这个例子中，play 许可要求每个用户交付\$AUD20 的费用（加上 10%的税）。

```
<play>
  <requirement>
    <peruse>
      <payment>
        <amount currency="AUD">20.00</amount>
        <taxpercent code="GST">10.0</taxpercent>
      </payment>
    </peruse>
  </requirement>
</play>
```

5. ODRL 条件模型

ODRL 支持权利条件的表示，Condition 说明当条件满足时相应的操作不再允许。ODRL 条件模型如图 6-14 所示^[5]。

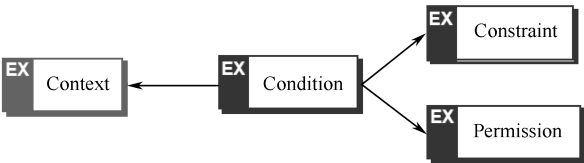


图 6-14 ODRL条件模型

条件 Condition 实体使用两个已经存在的实体。

- ① Permission: 定义了一个许可集来触发事件。
- ② Constraint: 定义了一个约束集来触发事件。

下面是 ODRL 条件模型的例子。这个例子中有两个许可：sell 和 play。play 许可有一个

软件类型的约束，当用这个软件播放视频的时候，play 许可必须终止。另外，有一个约束应用于所有的许可（play 和 sell），这个约束在空域（AU，澳大利亚），意味着如果许可在澳大利亚执行的话，许可必须终止。这说明不允许使用特定的某软件播放对象，以及在澳大利亚不允许播放和销售该对象。

```
<permission>
  <sell/>
  <play>
    <condition>
      <constraint>
        <software>...</software>
      </constraint>
    </condition>
  </play>
</permission>
<condition>
  <constraint>
    <spatial>
      <context>
        <uid>iso3166 :AU</uid>
      </context>
    </spatial>
  </constraint>
</condition>
```

6. ODRL 权限持有者模型

ODRL 支持对权限持有者的验证。权限持有者是可识别的，并且他们拥有对资源的使用权限。ODRL 权限持有者模型如图 6-15 所示^[5]。

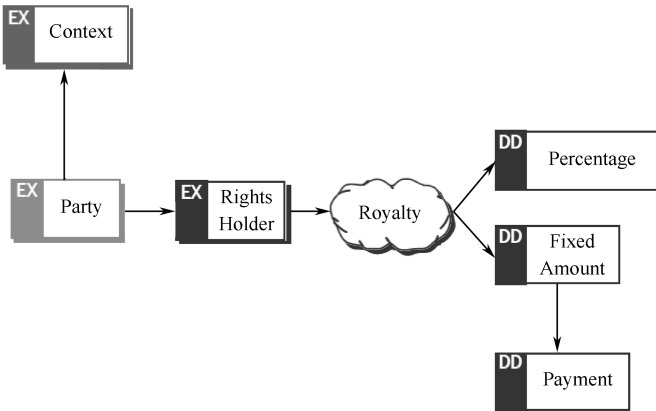


图 6-15 ODRL权限持有者模型

权限持有者实体包含了一个抽象实体（在图 6-20 中用云状图表示），抽象实体仅用于将类似的权限持有者的使用权限进行分类。

- ① Percentage（百分比）：表示对资源使用的交易费用的百分比。
- ② Fixed Amount（固定金额）：用固定值表示对资源使用的交易费用。

ODRL 的权限持有者模型可以用 XML 表示。下面的例子表示两个确定的用户对资源的使用比例分别为 90%和 10%。

```
<party>
  <context>...</context>
  <rightsholder>
    <percentage>90</percentage>
  </rightsholder>
</party>
<party>
  <context>...</context>
  <rightsholder>
    <percentage>10</percentage>
  </rightsholder>
</party>
```

7. ODRL 上下文模型

ODRL 支持表示实体和相关实体的附加信息。ODRL 上下文模型如图 6-16 所示^[5]。

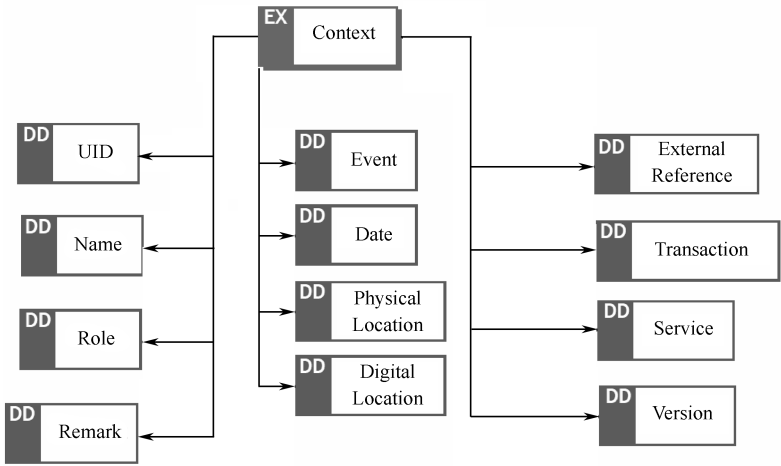


图 6-16 ODRL上下文模型

上下文实体包括了若干个其他的实体。

- ① UID：对实体的唯一标识码。
- ② Name：用于表示实体的名称。
- ③ Role：实体扮演的角色。
- ④ Remark：和实体相关的注释。

- ⑤ Version: 实体的版本。
- ⑥ Date: 实体产生或有效的日期。
- ⑦ Event: 事件的类型。
- ⑧ Physical Location: 事件/实体的物理位置。
- ⑨ Digital Location: 事件/实体的数字位置。
- ⑩ External Reference: 有关实体信息的链接。
- ⑪ Transaction: 和实体相关的购买交易的信息。
- ⑫ Service: 提供实体的服务链接。

上下文用于许多不同的目的，可和任何实体相关。在声明资源时，它用于表示资源的唯一标识符；当声明用户时，它表示用户的唯一标识符、他们可能扮演的角色，以及他们的名称；整个的权限（例如 Offer）在创建的时候可以用一个上下文作为唯一的标识符来表示；Agreement 实体可用上下文来提供交易的信息；基于文本的内容实体（即 Name 和 Remark）也可以用自然语言表示文本值（使用 XML 中的“lang”属性）。

下面的例子描述了一个用户的上下文。

```
<party>
  <context>
    <uid>x500:c=EX;o=FederalLibrary; ou=Registry; cn=MariaKBrown</uid>
    <name>Maria Brown</name>
    <role>onix:AO1</role>
    <reference>http://www.maria-k-brown.com/vcard.cml</reference>
  </context>
</party>
```

8. ODRL 提供者模型

ODRL 提供者模型如图 6-17 所示^[5]。

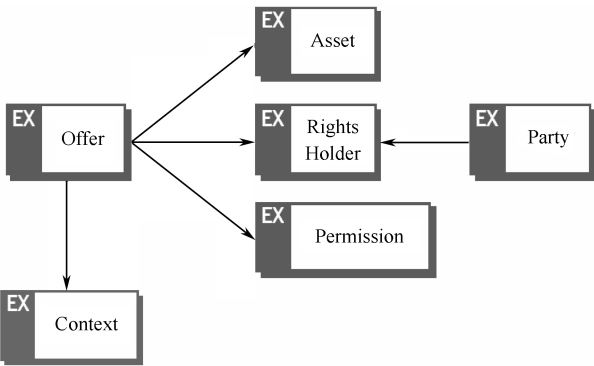


图 6-17 ODRL提供者模型

提供者实体包括如下内容。

- ① Asset: 资源的信息。
- ② Rights Holder: 提供者的信息。

- ③ **Permission:** 使用提供的权限信息（或链接）。
- ④ **Context:** 提供者的详细信息，如日期、时间、位置、标识符等。

Offer 实体允许表示特殊的权限持有者的详细信息，他们对其资源提供特定的权限。尽管不是强制的，但推荐使用上下文为 **Offer** 赋予唯一的标识符。一个 **Offer** 必须至少包含一个资源和权限；如果权限持有者（**Rights Holder**）没有指定，那么系统必须支持从别的地方提供这些信息。

ODRL 提供者模型可以用 **XML** 绑定来表示。下面的例子为资源权限用上下文描述了提供者。

```
<offer>
  <context>
    <uid>http://www.example.com/offer/38938238234723748888373</uid>
    <date><fixed>2001-10-10T09:00:00</fixed></date>
    <service>http://www.example.com/e-book-store</service>
  </context>
  <asset>...</asset>
  <party><permission>...</permission>
    <rightsholder>...</rightsholder>
  </party>
</offer>
```

9. ODRL 协议模型

ODRL 支持表示用户之间对于资源的特定权限的协议。**ODRL** 协议模型如图 6-18 所示^[5]。

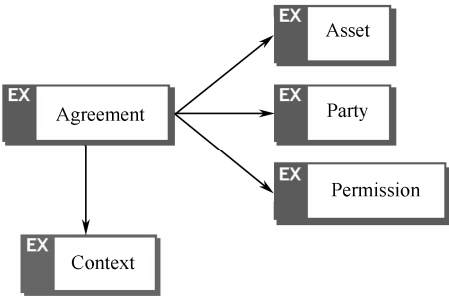


图 6-18 ODRL协议模型

协议（**Agreement**）实体包括如下内容。

- ① **Asset:** 资源的信息。
- ② **Party:** 协议的用户信息。
- ③ **Permission:** 同意使用权限的信息（或链接）。
- ④ **Context:** 协议的详细内容，如日期、时间、位置、标识符等。

协议实体允许表示特定的用户的详细信息，他们同意对某些资源具有特定的权限。尽管不是强制的，但推荐使用上下文为 **Agreement** 赋予唯一的标识符。一个 **Agreement** 必须至少包含一个资源和权限；如果用户（**Party**）没有指定，那么系统必须支持从别的地方提供这些信息。

ODRL 协议模型可以用 XML 绑定来表示。下面的例子用上下文描述了一个用户和资源权限集之间的协议。

```

<agreement>
  <context>
    <uid>doi:10.999/license/200110701/8736282828AAS</uid>
    <date><fixed>2001-07-01T10:31:30</fixed></date>
    <pLocation>Sydney, Australia</pLocation>
    <remark>Transacted by Example. Com</remark>
  </context>
  <party>
    <context>...</context>
  </party>
  <asset>...</asset>
  <permission>
    ...
  </permission>
</agreement>

```

10. ODRL 撤销模型

ODRL 支持撤销提供者、协议以及其他的权限表示。ODRL 撤销模型如图 6-19 所示^[5]。

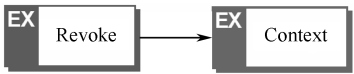


图 6-19 ODRL撤销模型

撤销实体包括 Context（标识符表示）。
 撤销实体允许通过上下文中的标识符表示撤销权限。标识符对系统来说必须是已知的。
 上下文中的唯一标识符可以用于：

- ① 所有的权限表示；
- ② 提供者（Offers）；
- ③ 协议（Agreements）；
- ④ 权限（Permissions）。

上面任何（或所有）的内容都可以撤销。在一个 Revoke 语句中，用多个上下文可以同时撤销多个表示。

ODRL 撤销模型可以用 XML 绑定来表示。下面的例子表示对协议的撤销，协议实体用 uid 元素表示。

```

<rights>
  <revoke>
    <context>
      <uid>doi:10.999/license/20010701/8736282828AAS</uid>
      <date><fixed>2001-10-30T12:30:30</fixed></date>
    </context>
  </revoke>
</rights>

```

```
<remark>Error in Original Agreement</remark>
</context>

</revoke>

</rights>
```

6.7.2 ODRL 安全模型

ODRL 支持两个安全权限表示：数字签名和指定资源的加密。安全模型用 W3C XML Signature [XML-SIG]和 W3C XML Encryption [XML-ENC] 规范进行描述，支持整个权限表示的封签（当使用 XML 绑定时），并且包括资源的加密信息。ODRL 安全框架用这两个规范中元素的子集来确保互操作性。

1. ODRL 加密模型

ODRL 加密模型如图 6-20 所示，包含了附加的 ODRL 实体（用“EX”标记的实体）、数字签名中规范的实体（用“DS”标记的实体）和加密规范中的实体（用“EC”标记的实体）^[5]。

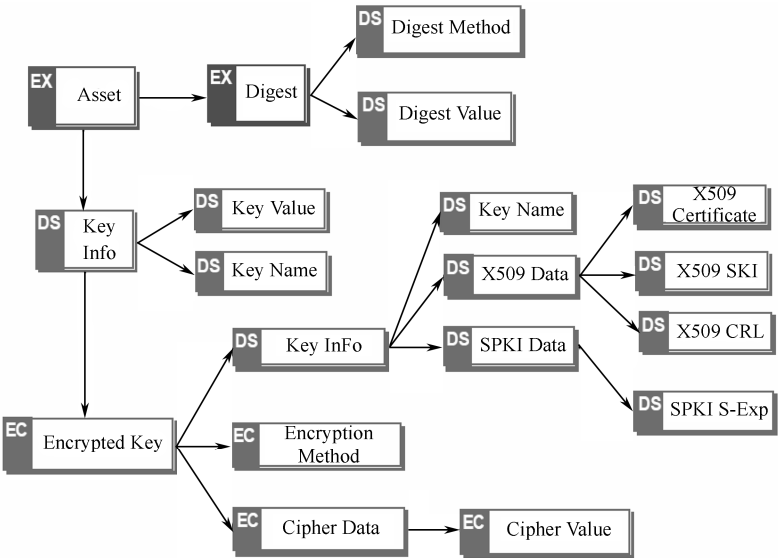


图 6-20 ODRL加密模型

为了表示资源加密的信息，ODRL 定义了一个新的实体“Digest”，它是 ODRL 资源(Asset)实体的孩子。Digest 实体用来保护和其他绑定的相关内容的完整性，包含以下子实体。

- ① Digest Method: 指定用于计算摘要值的算法，必须支持 SHA-1 算法。
- ② Digest Value: 所计算的摘要值。

Asset 实体包含一个“Key Info”实体，它可包括：

- ① “Key Value”和“Key Name”；
- ② 多个“Encrypted Key”子实体。

加密密钥（Encrypted Key）实体包括以下子实体。

- ① Encryption Method: 指定所用的加密算法，必须支持 RSA 算法。

② **Cipher Data**: 包括一个在 **CipherValue** 子实体中的待加密数据。

③ **Key Info**: 查看数字签名中关于这个实体的详细描述。

ODRL 加密框架有如下限制。

① 准许的加密方法（**Encryption Method**）为 **RSA**：

http://www.w3.org/2001/04/xmlenc#rsa-1_5

② 准许的 **Key Info** 为 **X509 Data**：

<http://www.w3.org/2000/09/xmldsig#X509Data>

2. ODRL 数字签名模型

ODRL 数字签名模型如图 6-21 所示，它包含数字签名规范中的实体（用“DS”标记的实体）和加密规范中的实体（用“EC”标记的实体）^[5]。

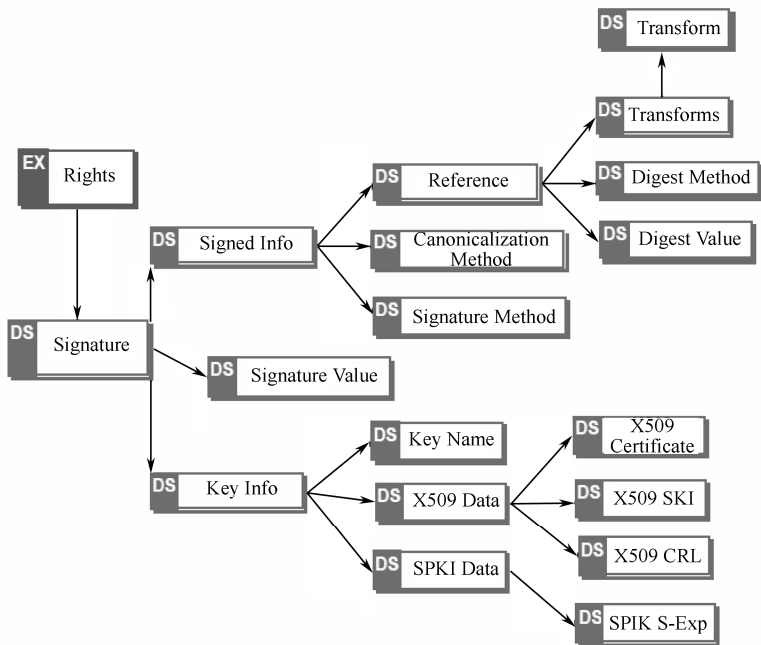


图 6-21 ODRL 数字签名模型

ODRL 利用 XML 签名规范的实体支持权限表示的数字签名，ODRL 表示和签名是在“Rights”实体中“封装”的。签名实体通过它的“id”属性和权限表示关联。

签名实体包括“Signed Info”实体，由下面三个实体组成。

① **Canonicalization Method**: 先用指定算法对 Signed Info 进行标准化，然后再完成签名计算，必须支持 C14N 算法。

① **Signature Method**: 指定用于产生签名的算法，必须支持 RSA 算法。

② **Reference**: 权限表示的链接（通过引用它的“id”）。引用实体还包含 Transform 实体，用来指定转换要求（“Enveloped Signature”和“C14N”必须按顺序）。引用实体还包含 Digest Method 和 Digest Value 实体。

签名实体还包括如下内容。

① **Signature Value:** Base64 编码签名值。

② **Key Info:** 这个实体包括三个子实体，“X509 Data”、“SPKI Data”和“Key Name”，“Key Name”实体包含一个字符串值，可被签名者用于向接收者传递一个密钥标识符；“SPKI Data”实体包含和 SPKI（Simple Public Key Infrastructure，简单的公钥基础设施）的公钥对、证书以及其他 SPKI 数据等相关的信息，并且包含“SPKI S-Exp”实体，它是一个 SPKI 典型的 S-表达式的 Base64 编码。

“X509 Data”实体包括以下内容。

① **X509 Certificate:** 包括一个进行了 Base64 编码的二进制 DER（Distinguished Encoding Rules，可辨别编码规则）X509 V3 证书。

② **X509 SKI:** 包括进行了 Base64 编码的 X509 V3 SKI（Subject Key Identifier，公钥标识）扩展的明文值（非 DER 编码）。

③ **X509 CRL:** 包括一个进行了 Base64 编码的 CRL（Certificate Revocation List，认证撤销列表）。

ODRL 数字签名框架是一个有效的签名（在[XML-SIG]中定义），受到如下的限制。

① 准许的 Canonicalization Method 是规范化形式的 XML：

<http://www.w3.org/TR/2001/REC-xml-c14n-20010315>

② 准许的 Signature Method 是 RSA：

<http://www.w3.org/2000/09/xmldsig#rsa-sha1>

③ 准许的 Digest Method 是 SHA-1：

<http://www.w3.org/2000/09/xmldsig#sha1>

④ 准许的 Transform 是封装的签名和规范化形式的 XML：

<http://www.w3.org/TR/2001/REC-xml-c14n-20010315>

<http://www.w3.org/2000/09/xmldsig#enveloped-signature>

⑤ 准许的 Key Info 是 X509 Data 和 SPK IData：

<http://www.w3.org/2000/09/xmldsig#X509Data>

<http://www.w3.org/2000/09/xmldsig#SPKIData>

3. 安全的 XML 实例

ODRL 安全模型可以用 XML 表示。下面的例子显示了资源 asset 元素包含加密，采用摘要和加密密钥。

```
<?xml version="1.0" encoding="UTF-8"?>
<o-ex:rights xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
             xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
             xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
             xmlns:enc="http://www.w3.org/2001/04/xmlenc#">
  <o-ex:context>
    <o-dd:uid>http://example.com/offers/3838383838.odrl</o-dd:uid>
    <o-dd:version>MRV Profile 2.8.99</o-dd:version>
  </o-ex:context>
```

```

<o-ex:offer>
  <o-ex:asset>
    <o-dd:uid>http://example.com/1793874932.mov</o-dd:uid>
  </o-ex:context>
  <o-ex:digest>
    <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <ds :DigestValue>--Base64-Encoded-Hash-Value--</ds :DigestValue>
  </o-ex :context>
  <ds :KeyInfo>
    <enc:EncryptedKey>
      <enc:CarriedKeyName>CEK-33247299234</enc:CarriedKeyName>
      <enc:EncryptionMethod>
        Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1U_5*/>
    <ds:KeyInfo>
      <ds:X509Data>
        <ds:X509SK1>-Base64-Encoded-Subject-Key-ID--</ds:X509SK1>
      </ds:X509Data>
      </ds:KeyInfo>
      <enc:CipherData>
        </enc:EncryptedKey>
      </ds:KeyInfo>
    </o-ex:asset>
  <o-ex:permission>
    <o-dd:play>
  </o-ex:permission>
</o-ex:offer>
</o-ex:rights>

```

下面的例子显示了一个进行了数字签名的权限表示。权限表示元素分配了“MyRightsData”的“id”。这个“id”属性和<context>元素中使用的“uid”不一样，尽管它们可以有相同的值，这个 id 在<Signature>元素中被<Reference>元素使用。<Transforms>元素指定了对 XML 表达式签名所要求的算法。

```

<o-ex:rights o-es:id="MyRightsData"
  xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
  xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <o-ex:context>
    <o-dd:uid>http://example.com/license/1191918237827.odrl</o-dd:uid>
  </o-ex:context>
  <o-ex:agreement>
    <o-ex:asset/>

```

```

        <o-ex:permission/>
        <o-ex:party>
    </o-ex:agreement>
    <ds:Signature>
        <ds:SignedInfo>
            <ds:CanonicalizationMethod
ds:Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
            <ds:Signature Method ds:Algorithm="http://www.w3.org/2000/09>xmldsig#rsa-sha1"/>
            <ds:Reference ds:URI="MyRights Data">
                <ds:Transforms>
                    <ds:Transform>
                        ds:Algorithm="http://www.w3.org/2000/09/mxldsig#enveloped-signature"/>
                    <ds:Transform
                        ds:Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-200110315"/>
                </ds:Transforms>
                <ds:DigestMethod ds:Algorithm="http://www.w3.org/2009/09/xmldsig#sha1"/>
                <ds:Digest Value>---base64-encode-hash-value---</ds:Signature Value>
            </ds:Reference>
        </ds:signedInfo>
    <ds:Signature Value>---base64-encoded-signature-value---</ds:Signature Value>
    <ds:KeyInfo>
        <ds:X509Data>
            <ds:X509Certificate>---base64-encoded-signer-certificate---</ds:X509Certificate>
        </ds:X509Data>
    </ds:KeyInfo>
</ds:Signature>
</o-ex:rights>

```

上面两个例子都需要包括一个混合的 XML 名称空间（支持 ODRL、XML 签名和 XML 加密），并且要在名称空间的前缀中明确表示。

在相同的环境中，有可能仅须对 ODRL 的部分内容进行加密，例如，对权限持有者的支付信息保密。上面描述的安全模型可以支持这种需求，特定的 ODRL 元素的加密可以参考 [XML-ENC]。

6.7.3 ODRL 表达式

1. 表达式容器

ODRL 支持三种将实体聚合成容器的关系^[5]。

- ① And 关系；
- ② Exclusive Or 关系，仅支持单一实体；

③ Inclusive Or 关系，支持单一或多个实体。

ODRL 容器结构可以用 XML 表示。下面的例子中，播放权限被限制为 CPU 或存储设备，并且要求播放前支付\$AUD200 或每次支付\$AUD1.50。

```
<permission>
  <play>
    <containet type="in-or">
      <cpu/>
      <storage/>
    </container>
  </play>
</Constraint>
<requirement>
  <container type="ex-or">
    <prepay>
      <payment>
        <amount currency="AUD"200.00</amount>
      </payment>
    </prepay>
    <peruse>
      <payment>
        <amount currency="AUD">1.50</amount>
      </payment>
    </peruse>
  </container>
</requirement>
</permission>
```

2. 表达式序列

ODRL 支持若干机制，可以指定实体的序列。一个实体的序列意味着实体是有顺序关系的，有两类序列^[5]。

① total: 顺序是绝对的。

② partial: 顺序是可选的。

一个新的结构，称为“sequence”，可用于聚集其他实体，以执行上述的语义。sequence 结构有一个“order”属性（具有固定值“total”和“partial”），用来指定序列的顺序，默认的序列顺序是“total”。另外，当没有 sequence 结构使用时，也就不需要假设顺序关系。在 sequence 结构中，顺序用“seq-item”元素指定，用属性指定序列号，属性“number”是大于等于 1 的整数。注意，顺序由这个 number 属性指定，而不是表达式的物理顺序。

sequence 的语义仅用于其直接孩子。

典型的 sequence 结构用来对权限、约束、条件和要求指定顺序，而不用于其他的实体。

ODRL 的 sequence 结构可以用 XML 表示。下面的例子中，对播放权限的要求进行绝对的排序：

- ① 在服务提供者那里注册用户的详细信息；
- ② 支付所需的费用。

```
<permission>
  <play>
    <requirement>
      <sequence order="total">
        <seq-item number="1">
          <register>
            <context>
              <service>http://example.com/registerhere</service>
            </context>
          </register>
        </seq-item>
        <seq-item number="2">
          <prepay>
            <payment>
              <amount currency="AUD">100.00</amount>
            </payment>
          </prepay>
        </seq-item>
      </sequence>
    </requirement>
  </permission>
```

3. 表达式连接

ODRL 表达语言支持表达式片段的连接。表达式片段的连接意味着支持明确的语义。连接允许 ODRL 表达式用存在的表达式进行构建，并且允许片段的显式连接。每个 ODRL 片段可以用“id”属性唯一指定，用“idref”属性进行引用。

通常情况下，在 ODRL 权限表达式中出现的实体假设是相关的。也就是说，一个在 asset 中出现的 permission 实体假设是相关的（即表示资源的权限）。

ODRL 连接结构可以用 XML 表示。下面的例子中，销售（sell）权限仅适用于 ASSET-01，而借贷（lend）权限可用于 ASSET-01 和 ASSET-02。

```
<rights>
  <offer>
    <asset id="ASSET-01">
      <context>...<context>
    </asset>
    <asset id="ASSET-02">
```



```

    <context>...</context>
  </asset>
  <permission>
    <asset idref="ASSET-01">
      <sell/>
    </permission>
  <permission>
    <asset idref="ASSET-01">
      <asset idref="ASSET-02">
        <lend>
      </asset>
    </permission>
  </offer>
</rights>

```

4. 表达式继承

ODRL 表达语言支持资源之间的表达式继承，也就是说，允许定义父/子关系。

子资源元素可以包含 “inherit” 元素，指定从其他的父资源继承权限。所有的父资源权限将和子资源权限合并。在这个合并过程中，结果权限表达式不允许发生冲突。

继承元素 “inherit” 对子资源有一个 “override” 属性，即如果其值为 true，将不能继承父资源权限，但可以定义它自己的权限，默认值为 false。

对父资源，inherit 元素也有一个属性值 “default”，即如果这个值为 true，将不允许子资源重载它的权限，默认值为 false。

“override” 和 “default” 不能同时为 true。

ODRL 继承结构可以用 XML 表示。下面的例子中，第一个权限表达式为可识别的资源指定了播放权限；第二个权限表达式，对于和第一个相关的资源，指定了从可识别的资源继承而来的权限。

表达式还指定了不对继承的资源进行重载（默认值），因此，对第二个资源的完整权限包括播放和授予权限。如果第二个表达式指定了对继承的资源进行重载，那么对第二个资源就只有授予的权限。

```

<rights>
  <offer>
    <asset>
      <context>
        <uid>urn:example:asset:007</uid>
      </context>
    </asset>
    <permission>
      <play/>
    </permission>
  </offer>

```

```
</rights>

<rights>
  <offer>
    <asset>
      <context>
        <uid>urn:example:asset:007-Part1</uid>
      </context>
      <inherit override="false"default="false">
        <context>
          <uid>urn:example:asset:007</uid>
        </context>
      </inherit>
    </asset>
    <permission>
      <give/>
    </permission>
  </offer>
</rights>
```

6.7.4 ODRL XML 语法

ODRL 是一个有效的 XML 语法，这种语法通过 XML Schema 正式定义。这些是 ODRL 表达语言的规范引用和数据字典。

提示：如果人类的自然语言需要对包含字符串的任何元素进行规范，那么推荐使用包含 XML 名称空间的标准的“xml:lang”属性。

1. ODRL XML Schema

ODRL 使用两个 XML Schema。一个构架定义了表达式语言的元素和结构，另一个定义了数据字典的元素。两者都必须有效的 ODRL 表达式，此外数据字典构架依赖于表达式语言构架，如前一节表达式语言模型所定义的元素。

2. ODRL XML 名称空间

ODRL 支持用 XML 名称空间表示它的元素和其他内容描述元素的范围和特性。
ODRL 表达式语言的 XML 名称空间 URI 1.1 版本：

```
http://odrl.net/1.1/ODRL-EX
```

ODRL 数据字典的 XML 名称空间 URI 1.1 版本：

```
http://odrl.net/1.1/ODRL-DD
```

注意：在 ODRL 规范被形式化描述，以及新的 XML 名称空间和 URI 被确定之前，应该考虑到这些 URI 的实验性。

3. ODRL 连接

ODRL 使用 XML Schema 的 ID 和 IDREF 在 XML 片段引用其他片段，这被用来表示核心的 ODRL 实体之间的关系，如 Asset、Permission、Offer、Agreement 之间的关系，这些元素可以用“id”属性来识别，并通过“idref”属性引用。

重要的是要认识到，ODRL 表达式变得更复杂，为了具有可管理和可重用的权限表达式，连接的划分和表示就变得很重要。连接机制能生成相当复杂的表达式，同时保留整个权限语言的可读性。

6.7.5 ODRL XML 例子

XML 语法通过一系列的方案，涵盖了不同的内容（电子书、视频、教育等）。下面的例子摘自文献[5]，注意例子中所用的一些 XML 命名空间（词汇值）是虚构的。

1. 电子书方案 1

Corky Rossi（作者）和 Addison Rossi（插图作者）通过“EBooksRUS Publishers”（出版社）出版他们的电子书。他们希望消费者购买电子书（要求支付\$AUD20.00 加 10%的税），限制在单个的 CPU，并最多允许打印 2 份拷贝。他们还允许在线免费浏览电子书的前 5 页。注意使用 ONIX 命名空间的 NumberOfPages 实体表示单位类型。

收入分配的 60%归作者，10%归插图作者，30%归出版社。他们的身份可在一个 X.500 库中识别，他们的角色用 ONIX 和 MARC 指定。

这个方案在 ODRL 中的 XML 编码如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<o-ex:rights xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
  xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
  xmlns:onix="http://www.editeur.org/onix/ReferenceNames"
  xmlns:marc="http://www.loc.gov/marc/">
  <o-ex:offer>
  <o-ex:asset>
  <o-ex:context>
  <o-dd:uid>urn:ebook.world/999999/ebook/rossi-000001</o-dd:uid>
    <o-dd:name>Why Cats Sleep and We Don't</o-dd:name>
  </o-ex:context>
  </o-ex:asset>
  <o-ex:permission>
    <o-dd:display>
```

```

<o-ex:constraint>
    <o-dd:cpu/>
</o-ex:constraint>
</o-dd:display>
<o-dd:print>
<o-ex:constraint>
<o-dd:count>2</o-dd:count> </o-ex:constraint>
</o-dd:print>
<o-ex:requirement>
    <o-dd:prepay>
<o-dd:payment>
<o-dd:amount o-dd:currency="AUD">20.00</o-dd:amount>
    <o-dd:taxpercent o-dd:code="GST">10.00</o-dd:taxpercent> </o-dd:payment>
</o-dd:prepay>
    </o-ex:requirement> </o-ex:permission> <o-ex:permission>
    <o-dd:display>
<o-ex:constraint>
<o-dd:unit o-ex:type="onix:NumberOfPages">
    <o-ex:constraint>
<o-dd:range>
    <o-dd:min>1</o-dd:min>
    <o-dd:max>5</o-dd:max>
</o-dd:range>
</o-ex:constraint> </o-dd:unit>
</o-ex:constraint>
    </o-dd:display>
</o-ex:permission>
<o-ex:party>
<o-ex:context>
<o-dd:uid>x500:c=AU;o=RightsDir;cn=CorkyRossi</o-dd:uid>
<o-dd:role>onix:roles/A01</o-dd:role>
</o-ex:context>
<o-ex:rightsholder>
<o-dd:percentage>60</o-dd:percentage> </o-ex:rightsholder>
</o-ex:party>
<o-ex:party>
<o-ex:context>
<o-dd:uid>x500:c=AU;o=RightsDir;cn=AddisonRossi</o-dd:uid>
<o-dd:role>onix:roles/A12</o-dd:role>

```

```

</o-ex:context>
<o-ex:rightsholder>
<o-dd:percentage>10</o-dd:percentage> </o-ex:rightsholder>
</o-ex:party>
<o-ex:party>
<o-ex:context>
<o-dd:uid>x500:c=AU;o=RightsDir;cn=EBooksRUS</o-dd:uid>
<o-dd:role>marc:roles/pbl</o-dd:role>
</o-ex:context>
<o-ex:rightsholder>
<o-dd:percentage>30</o-dd:percentage> </o-ex:rightsholder>
</o-ex:party>
</o-ex:offer>
</o-ex:rights>

```

2. 电子书方案 2

继上面的电子书方案 1，消费者 Mary Smith 决定购买电子书，ODRL 表达式显示了生成的协议。协议有一个上下文（具有标识符和日期），权限显示了对消费者的详细约束。在这种情况下，CPU 标识符和许可证一起保存。

```

<?xml version="1.0" encoding="UTF-8"?>
<o-ex:rights xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
  xmlns:o-dd="http://odrl.net/1.1/ODRL-DD">
  <o-ex:agreement>
  <o-ex:context>
  <o-dd:uid>urn:ebook.world/999999/license/1234567890-ABCDEF</o-dd:uid>
  <o-dd:pLocation>Sydney, Australia</o-dd:pLocation>
  <o-dd:remark>Transacted by Example.Com</o-dd:remark> </o-ex:context>
  <o-ex:asset>
  <o-ex:context>
  <o-dd:uid>urn:ebook.world/999999/ebook/rossi-000001</o-dd:uid> </o-ex:context>
  </o-ex:asset>
  <o-ex:permission>
  <o-dd:display>
  <o-ex:constraint>
  <o-dd:cpu>
  <o-ex:context>
  <o-dd:uid>Adobe-WebBuy:CPD-ID:ER-393939-DSS-787878</o-dd:uid> </o-ex:context>
  </o-dd:cpu>
  </o-ex:constraint>

```

```

</o-dd:display>
<o-dd:print>
<o-ex:constraint>
<o-dd:count>2</o-dd:count> </o-ex:constraint>
</o-dd:print>
<o-ex:requirement>
  <o-dd:prepay>
<o-dd:payment>
<o-dd:amount o-dd:currency="AUD">20.00</o-dd:amount>
  <o-dd:taxpercent o-dd:code="GST">10.00</o-dd:taxpercent>  </o-dd:payment>
</o-dd:prepay>
  </o-ex:requirement>  </o-ex:permission> <o-ex:party>
<o-ex:context>
<o-dd:uid>urn:ebook.world/999999/users/msmth-000111</o-dd:uid> <o-dd:name>Mary Smith</o-dd:name>
</o-ex:context>
</o-ex:party>
</o-ex:agreement> </o-ex:rights>

```

3. 电子书方案 3

电子书将一个“电子图书交换”的电子书凭证的标题命名为“XML: A Manager’s Guide”，权限拥有者是 Addison-Wesley。分发商（一家名为“XYZ”的公司）对这本书有一个协议，他们有权将这本书卖到 5000 份复制。这本书的另一个终端用户 John Doe 有一个许可证，他所有的权限从 2001 年年初开始，到 2004 年年底结束。他有权浏览此书共 30 天；他最多可在“受信任的打印机”上打印 5 份复制；他每周可在任何打印机上打印第 1~100 页之间的 5 页，总共最多可打印 100 页；他也可以每周提取 5000 字节到剪贴板（内存），总共最多可提取 1,000,000 字节；他有权在许可开始一年后把书的所有权转让。

注意 ONIX 和 EBX 名称空间的 NumberOfPages 和 NumbeOfBytes 实体的使用，以及 id 和 idref 的使用，它们指定了协议/权限持有者和资源之间的关系。

这个方案在 ORDL 中的 XML 编码如下：

```

<?xml version="1.0" encoding="UTF-8"?>
<o-ex:rights xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
  xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
  xmlns:onix="http://www.editeur.org/onix/ReferenceNames"
  xmlns:ebx="http://www.ebxwg.org/ebook/vocab/"
  xmlns:marc="http://www.loc.gov/marc/">
  <o-ex:context>
    <o-dd:uid>urn:ebook.world/999999/voucher/2001/1234567890</o-dd:uid> <o-dd:date>
    <o-dd:fixed>2001-05-01T08:30:00</o-dd:fixed> </o-dd:date>
    <o-dd:event>issued</o-dd:event> </o-ex:context>

```

```

<o-ex:asset o-ex:id="a001">
  <o-ex:context>
<o-dd:uid>isbn:872-2345-981</o-dd:uid>
<o-dd:name>XML: A Manager's Guide</o-dd:name> </o-ex:context>
</o-ex:asset>
<o-ex:party>
<o-ex:context>
<o-dd:uid>http://publishers.net/registry/AWL </o-dd:uid>
<o-dd:name>Addison-Wesley </o-dd:name>
<o-dd:reference>http://www.addison-wesley.com</o-dd:reference>
</o-ex:context>
<o-ex:rightsholder/> </o-ex:party>
<o-ex:agreement>
<o-ex:asset o-ex:idref="a001"/> <o-ex:party>
<o-ex:context>
<o-dd:uid>http://distributors.net/registry/xyz</o-dd:uid>
  <o-dd:name>XYZ Company </o-dd:name>
  <o-dd:role>marc:dst</o-dd:role>
</o-ex:context>
</o-ex:party>
<o-ex:permission>
  <o-dd:sell>
<o-ex:constraint>
<o-dd:count>5000</o-dd:count> </o-ex:constraint>
</o-dd:sell>
</o-ex:permission>
</o-ex:agreement>
<o-ex:agreement>
<o-ex:asset o-ex:idref="a001"/> <o-ex:party>
<o-ex:context>
<o-dd:uid>http://people.net/registry/john-doe-9999</o-dd:uid>
<o-dd:name>John Doe</o-dd:name>
</o-ex:context>
</o-ex:party>
<o-ex:permission>
  <o-ex:constraint>
<o-dd:datetime>
<o-dd:start>2001-01-01T00:00:00</o-dd:start>
  <o-dd:end>2004-12-31T23:59:59</o-dd:end> </o-dd:datetime>

```

```

</o-ex:constraint>
<o-dd:display>
<o-ex:constraint>
<o-dd:accumulated>P30D</o-dd:accumulated> </o-ex:constraint>
</o-dd:display>
<o-dd:print>
<o-ex:container o-ex:type="and">
    <o-ex:constraint>
<o-dd:count>5</o-dd:count> <o-dd:printer>
<o-ex:context>
<o-dd:uid>guid:TrustPrint/4747474742222</o-dd:uid> </o-ex:context>
</o-dd:printer>
</o-ex:constraint>
<o-ex:constraint>
<o-dd:unit o-ex:type="onix:NumberOfPages">
    <o-ex:constraint>
<o-dd:count>100</o-dd:count> </o-ex:constraint>
</o-dd:unit>
<o-dd:printer/>
</o-ex:constraint>
<o-ex:constraint>
<o-dd:unit o-ex:type="onix:NumberOfPages">
    <o-ex:constraint>
<o-dd:range>
<o-dd:min>1</o-dd:min>
<o-dd:max>100</o-dd:max> </o-dd:range>
<o-dd:count>5</o-dd:count> </o-ex:constraint>
</o-dd:unit>
<o-dd:interval>P7D</o-dd:interval> <o-dd:printer/>
</o-ex:constraint>
    </o-ex:container> </o-dd:print>
<o-dd:excerpt>
<o-ex:constraint>
    <o-dd:memory>
<o-ex:container o-ex:type="and">
    <o-ex:constraint>
<o-dd:unit o-ex:type="ebx:NumberOfBytes">
    <o-ex:constraint>
<o-dd:count>1000000</o-dd:count> </o-ex:constraint>

```



```

</o-dd:unit>
</o-ex:constraint>
<o-ex:constraint>
<o-dd:unit o-ex:type="ebx:NumberOfBytes">
  <o-ex:constraint>
<o-dd:count>5000</o-dd:count>
<o-dd:interval>P7D</o-dd:interval>
</o-ex:constraint> </o-dd:unit>
</o-ex:constraint>
  </o-ex:container>
  </o-dd:memory>
  </o-ex:constraint>
</o-dd:excerpt>
<o-dd:give>
<o-ex:constraint>
<o-dd:datetime>
<o-dd:start>2002-01-01T00:00:00</o-dd:start> </o-dd:datetime>
</o-ex:constraint>
</o-dd:give>
</o-ex:permission>
</o-ex:agreement>
</o-ex:rights>

```

4. 视频方案 1

一个视频有三个权利持有者，Massimo Canale 收到 75% 的交易，Simona Canale 收到另外的 25%，此外 Maria Canale 收到 Simona Canale 10% 的份额。请注意，最后两方是嵌套的。

视频提供两个不同层次的质量（30 和 90dpi），价格不等，按每次使用（peruse）收费。

注意使用 mpeg7 名称空间的“resolution”实体的使用，它指定了资产的质量。另外，整个表达式有一个具有唯一标识符的上下文。

这个方案在 ORDL 中的 XML 编码如下：

```

<?xml version="1.0" encoding="UTF-8"?>
<o-ex:rights xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
  xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
  xmlns:mpeg7="http://www.mpeg7.org/2001/MPEG-7_Schema">
<o-ex:context>
<o-dd:uid>doi:voucher/383838383</o-dd:uid>
<o-dd:name>The Voucher for XML: The Movie</o-dd:name>
<o-dd:dLocation>http://example.com/odrl/383838383.xml</o-dd:dLocation>
</o-ex:context>

```

```

<o-ex:offer>
<o-ex:asset>
<o-ex:context>
<o-dd:uid>doi:0.9999999/video/383838383</o-dd:uid>
  <o-dd:name>XML: The Movie</o-dd:name>
</o-ex:context>
</o-ex:asset>
<o-ex:party>
<o-ex:context>
<o-dd:uid>x500:c=IT;o=Registry;cn=MassimoCanale</o-dd:uid> </o-ex:context>
<o-ex:rightsholder>
<o-dd:percentage>75</o-dd:percentage> </o-ex:rightsholder>
</o-ex:party>
<o-ex:party>
<o-ex:context>
<o-dd:uid>x500:c=IT;o=Registry;cn=SimonaCanale</o-dd:uid> </o-ex:context>
<o-ex:rightsholder>
<o-dd:percentage>25</o-dd:percentage> </o-ex:rightsholder>
<o-ex:party>
<o-ex:context>
<o-dd:uid>x500:c=IT;o=Registry;cn=MariaCanale</o-dd:uid> </o-ex:context>
<o-ex:rightsholder>
<o-dd:percentage>10</o-dd:percentage> </o-ex:rightsholder>
</o-ex:party>
</o-ex:party>
<o-ex:permission>
  <o-dd:play>
<o-ex:constraint>
<o-dd:quality o-ex:type="mpeg7:resolution">
  <o-ex:constraint>
<o-dd:range>
<o-dd:max>30</o-dd:max> </o-dd:range>
</o-ex:constraint>
  </o-dd:quality>
</o-ex:constraint>
<o-ex:requirement>
  <o-dd:peruse>
<o-dd:payment>
<o-dd:amount o-dd:currency="ITL">1000.00</o-dd:amount> </o-dd:payment>
</o-dd:peruse>
  </o-ex:requirement> </o-dd:play>

```

```

<o-dd:play>
<o-ex:constraint>
<o-dd:quality o-ex:type="mpeg7:resolution">
  <o-ex:constraint>
<o-dd:range>
<o-dd:max>90.0</o-dd:max> </o-dd:range>
</o-ex:constraint>
  </o-dd:quality>
</o-ex:constraint>
<o-ex:requirement>
  <o-dd:peruse>
<o-dd:payment>
<o-dd:amount o-dd:currency="ITL">5000.00</o-dd:amount> </o-dd:payment>
</o-dd:peruse>
  </o-ex:requirement> </o-dd:play>
</o-ex:permission>
</o-ex:offer>
</o-ex:rights>

```

5. 超级分发实例 1

对一个特定的个体 (JJJones), 公司提供免费的视频资源 (如上例的视频方案 1) 分发 (即转让权限), 截止日期为 2001 年 12 月 31 日。他们通过允许权限表达式 (其本身是一个资源) 的所有权转让来实现。

请注意, 这种情况下视频不是资源, 而是把权限表达式当成了资源。

这个方案在 ORDL 中的 XML 编码如下:

```

<?xml version="1.0" encoding="UTF-8"?>
<o-ex:rights xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
  xmlns:o-dd="http://odrl.net/1.1/ODRL-DD">
  <o-ex:agreement>
  <o-ex:asset>
  <o-ex:context>
  <o-dd:uid>doi:10.9999999/voucher/383838383</o-dd:uid>
    <o-dd:name>The Voucher for XML: The Movie</o-dd:name> </o-ex:context>
  </o-ex:asset>
  <o-ex:party>
  <o-ex:context>
  <o-dd:uid>x500:c=US;o=Example;cn=JJJones</o-dd:uid> </o-ex:context>
  </o-ex:party>
  <o-ex:permission>
  <o-dd:give>

```

```
<o-ex:constraint>
<o-dd:datetime><o-dd:end>2001-12-31T23:59:59</o-dd:end>
</o-dd:datetime> </o-ex:constraint>
</o-dd:give>
</o-ex:permission>
</o-ex:agreement>
</o-ex:rights>
```

6.8 LicenseScript 简介

6.8.1 基于 XML 的权限描述语言存在的问题

前面介绍了基于 XML 的权限描述语言——XrML 和 ODRL。但基于 XML 的权限描述语言存在一些缺点：

- ① 当条件变得复杂的时候，语法变得复杂、晦涩；
- ② 语言缺少形式语义，不能进行逻辑推理，许可证的解释依赖于语法和直观语义；
- ③ 语言不能描述版权法规。

针对这些问题，国外学者 Gunter 和 Pucella 等人^{[13][14]}希望在数字权利的动态描述领域取得突破，于是他们在数字权限的形式语义和逻辑推理方面都进行了深入的研究。Gunter 借用了程序语言语义技术来定义许可证（license）的语义，认为一个许可证的语义就是动作执行序列（trace）的集合，每个 trace 表示该许可证允许的动作序列。一个正确的执行过程就是允许许可证定义的合法动作序列得到执行，而禁止其他序列执行。Pucella 等人则在 Gunter 的基础上，进一步尝试了采用一种称为“lic”的逻辑来推理数字权限^[15]。

总的来说，他们对数字权利的动态描述主要集中在两大主线。

（1）利用许可证定义用户合法的动作执行序列

通过对用户动作执行序列的跟踪，记录许可证的动态演化过程。通常的做法是利用程序设计语言技术来定义许可证的语义。许可证详细地定义了合法动作执行序列，而禁止其他动作序列的执行^[16]。这种许可证的动作执行序列的定义方式，必须对数字资源的执行设备、执行时间、执行环境、执行序列等都做详细事先定义，从结构上来讲是完备的，从理论上来说也是可行的，但是如果深入推敲这种处理方法，不难发现这种手段其实有很多缺陷。因为在实际应用过程中，用户使用数字资源的动作序列是一个随机、灵活的过程，这个过程与当时用户的需求、动机、外部设备状况等诸多主客观因素都有千丝万缕的联系。事先定义好所有的用户对数字资源的主观执行序列是不可行的^[17]。

（2）利用逻辑推理来实现证书的动态演化

逻辑推理并不需要对许可证的演化进行事先的定义，而且在执行过程中，根据用户的访问请求可以自动执行数字证书的自动推理，在相应的逻辑推理规则的约束下，自动完成数字权利的动态描述。在克服第一大研究主线缺陷的同时，也符合使用控制模型中对数字资源权

利使用的连续性、易变性、随机性和灵活性等特性的要求^{[17][18]}。

在文献[1]中,荷兰学者 Cheun Ngen Chong 等人提出了基于多重集 (multi-set) 和逻辑编程的权利描述语言——LicenseScript。该语言与其他权利表达语言的不同之处在于它分成静态部分和动态部分。关于内容的术语和使用条件构成静态部分,这些术语和使用条件一般都是按照法律、规章和商业规则产生的,用 Prolog 的子句表达。Cheun Ngen Chong 又提出了由于许可证是在不断变化的上下文中使用的,因此许可证也必须具有发展变化的能力。所以 LicenseScript 中的动态部分将许可证看成某个多重集的一个元素,在这个多重集中可以应用重写规则。这些规则描绘了上下文 (设备和系统) 按照许可证行事的方式。这样一个许可证就有了双重性 (静态和动态),这两个层次是由一组表现当前状态的绑定 (bindings) 联系起来的^[19]。

LicenseScript 描述了三个角色:信息提供者 (例如内容提供者、个人向网站提供的个人数据等)、信息消费者 (例如消费者、公司职员等)、信息监控者 (例如入侵检测)。

LicenseScript 由两个模块组成:许可证和规则。

6.8.2 许可证

在 LicenseScript 中,许可证定义了对于数字内容使用的条件,因此至少包含两方面相关的信息^[1]:

- ① 被授权数据的引用;
- ② 这些数据的使用条件。

在 LicenseScript 中,许可证用 $\text{lic}(\text{content}, \Delta, B)$ 形式表示^{[1][20]}。其中:

- ① content 是一个唯一的标识符,表示许可证引用的数据。
- ② Δ 是一个子句的集合,即 Prolog 程序,定义了某个操作 (如播放) 何时允许执行。
- ③ B 是一个绑定 (bindings) 集,即一个包含形如 $\text{name} \equiv \text{value}$ 的元素的集合。例如, $\{\text{expires} \equiv 10/10/2003\}$ 是只有一个元素的绑定集。

绑定集提供了一种灵活的方式来存储修改的数据。许可证可认为是一个数据库, Δ 是其内核部分, B 是其扩展。为了使许可证与外部进行交互,定义一组保留的调用,构成许可证的“API”。在定义精确的 LicenseScript 可操作语义之前 (包括多重集重写规则),让我们对子句和绑定集如何关联先有些感性的知识。例如, $\text{canplay}(\cdot)$ 表示许可证何时允许播放所给的音乐片段:如果程序 Δ 中查询 $\text{canplay}(B, B')$ 成功,则意味着许可证 $\text{lic}(a, \Delta, B)$ 允许播放音乐片段 a 。注意,绑定集 B 是作为一个查询的参数,在 $\text{canplay}(B, B')$ 成功后, B' 将包含新的许可证绑定。下面介绍几个许可证的例子。

- ① 许可证 $\text{lic}(\text{mus}, \{\text{canplay}(X, X) : -\text{true}\}, \{\})$ 表示可以播放 mus 。
- ② 许可证 $\text{lic}(\text{mus}, \{\}, \{\})$ 表示不允许对 mus 进行任何操作。
- ③ 许可证 $\text{lic}(a, \Delta, \{\text{expire} \equiv 10/10/2003\})$, 这里 Δ 包含一个子句:

```
{canplay(B, B) :- today(D),
    get_value(B, expire, Exdate),
    Exdate > D;}
```

许可证表示允许在所给的截止日期之前演奏 a 。 $\text{today}(D)$ 和 $\text{get_value}(B, n, V)$ 是 API 中的两个原语， $\text{today}(D)$ 绑定变量 D 到现在的系统日期， $\text{get_value}(B, n, V)$ 根据绑定集 B 将 n 的值返回到 V 中，将这些原语放在一个称为域的特定程序中。许可证可以放到很多域中，不同的域原语可以有不同的含义。

④ 某些情况下，许可证的执行可能会对绑定集进行修改。考虑一个许可证的实例，在给定的次数内允许播放（play）一个音乐片段：每次执行 play 操作，计数器加 1，由原语 $\text{set_value}(\text{Old}B, \text{name}, \text{value}, \text{New}B)$ 来实现。这个原语允许在新的绑定 $\text{New}B$ 中 name 和一个新的值 value 相关联，用这种方法实现许可证的更新。现在考虑许可证： $\text{lic}(a, \Delta, \{\text{played_times} \equiv 3\})$ ，其中 Δ 包含下列子句：

$$\begin{aligned} \text{canplay}(B, B') : & \neg \text{get_value}(B, \text{played_times}, R), \\ & R < 10, \\ & \text{set_value}(\text{played_times}, R + 1, B') \end{aligned}$$

这里，首先提取变量 played_times 的值放到局部变量 R 中，如果这个值小于 10，那么将 played_times 加 1，并且将这个新值存储到绑定集 B' 中，这个绑定集除了 played_times ，其他和 B 相同。

6.8.3 重写规则

到目前为止，我们了解了许可证的结构，许可证所构成的多重集通常驻留在一个设备里，那么许可证是如何与外部通信的呢？

设备和许可证之间的通信模型由多重集重写规则来完成，这些规则可认为是设备中的固件。直观地说，当许可证发生变化时（它们可能会被获取或删除），设备中的规则是固定的（当然规则有时也可以更新）。下面给出重写规则的定义。

☒ 定义 6-1 重写规则是一个 4 元组 $\text{rule}(\text{arg}): \text{lms} \rightarrow \text{rms} \Leftarrow \text{cond}$

这里 $\text{rule}(\text{arg})$ 是一个称为规则标签（rule label）原子， lms 和 rms 是两个多重集， cond 是形如 $P_i \vdash Q_i$ 的元素序列。

下面是一个规则的例子：

$$\begin{aligned} \text{play}(X) : & \text{lic}(X, \Delta, B) \rightarrow \text{lic}(X, \Delta, B') \\ & \Leftarrow \Delta \vdash \text{canplay}(B, B') \end{aligned}$$

规则的左部 $\text{lic}(X, \Delta, B)$ 首先和 MS 中的一个许可证 lic 进行匹配（这个实例中， Δ 和 B 分别是 Prolog 程序和许可证绑定），如果没有发现可匹配的许可证，操作失败。如果存在可匹配的许可证，则在程序 Δ 中查询 $\text{canplay}(B, B')$ ，若查询成功则 $\text{play}(X)$ 成功，并且在多重集 MS 中用 $\text{lic}(X, \Delta, B')$ 替代 $\text{lic}(X, \Delta, B)$ 。

更一般地，当规则 $\text{rule}(\text{arg}): \text{lms} \rightarrow \text{rms} \Leftarrow \text{cond}$ 启动时，首先将左多重集 lms 和 MS 的一个子多重集匹配；然后，执行 cond 中所有的查询，如果成功，那么操作成功，并且在 MS 中用右多重集 rms 替代 lms 。

我们再看一个例子。Amanda 从 Ben 那里买了一本书“ebook:prolog”，Ben 发放了一个许可证并将过期时间定为“23/06/2004”。

```

license(ebook:prolog,
  [(canprint(B1,B2,User):-
    get_value(B1,consumer,C),
    C=User,
    get_value(B1,expires,Exp),
    today(D),D<Exp,
    get_value(B1,printed,P),
    get_value(B1,max_prints,Max),
    P<Max,
    set_value(B1,printed,P+1,B2)],
  [(company=Ben),
  (consumer=Amanda),
  (expires=23/06/2004),
  (max_prints=2),
  (printed=0)])}

```

Amanda 在绑定部分符合条件的情况下可以得到 canprint 的操作权限。下面是 LicenseScript 的规则实例。

```

print(Ebook,User):
  license(Ebook,CLAUSE,B1)->
  license(Ebook, CLAUSE,B2)
  <= CLAUSE |-canprint(B1,B2,User)

```

在规则中，如果用户要求打印（print），就会先检查用户是否有 canprint 的推论结果，若有则将 license 更新，即将 license 的绑定从 B1 变成 B2。

大概的流程是，Amanda 先发送打印（print）的操作请求，查询规则，然后去查询许可证 license，得到 canprint 的结果；经过绑定集的重写，让 printed 这个变量的值从 0 变为 1，更新绑定集，完成 license 的转换。

6.8.4 LicenseScript 执行模型

在这一节中，形式化地定义规则的执行^[20]。

☒ 定义 6-2 匹配

替换 θ 记为 $\text{Dom}(\theta) = \{x \mid \theta(x) \neq x\}$ ，在 o 中出现的变量集合记为 $\text{Var}(o)$ 。如果 $t\sigma = s$ 且 $\text{Dom}(\sigma) = \text{Var}(t)$ ，则替换 σ 称为项 t 和 s 的匹配替换。这种情况下，称 t 匹配 s 。如果一个项和另一个项匹配，则存在一个唯一的匹配替换。

如前所述，许可证表示为 $\text{lic}(\text{content}, \Delta, B)$ ，假设所有可用的许可证存储在多重集 MS 中。LicenseScript 操作执行的整个过程如下。

① 用户向一个设备发送请求，要执行给定的操作，并给出了相关的信息。请求由 Prolog 原子表示，例如，`play(music_piece)`。

② 接收请求的设备，检查是否存在一个规则，其标签和请求匹配。例如， $\text{play}(a)$ 操作和规则 $\text{play}(X): \text{lic}(X, \Delta, B) \rightarrow \text{lic}(X, \Delta, B') \Leftarrow \Delta \vdash \text{canplay}(B, B')$ 的头部匹配。在这种情况下匹配替换是 $\sigma = \{X/a\}$ 。

③ 假设规则 $\text{rule}(\text{arg}): \text{lms} \rightarrow \text{rms} \Leftarrow \text{cond}$ 和原子请求匹配，其匹配实例为 σ_1 。检查在 MS 中是否存在一个（或多个）许可证可以和规则的左侧匹配，如果存在 MS 的一个子多重集 lics 和替换 σ_2 ，使得：

$$\text{lms}\sigma_1\sigma_2 = \text{lics}$$

并且 $\text{cond}\sigma_1\sigma_2$ 成功，替换结果为 σ_3 。那么，请求的操作就被授权执行，返回的 σ_3 就执行新的绑定。实际上，这里可能存在某种不确定的机制，既然可能有不同的 MS 的子多重集 lics ，满足上述的条件，甚至有多于一个的 σ_2 。这对应于用户有多个许可证的情况，允许选择想要的操作。在这种情况下，可以假设系统向用户提问使用哪个许可证。

④ 最后一步是更新多重集：用 $\text{rms}\sigma_1\sigma_2\sigma_3$ 置换 MS 中的 lics 。

再看一个例子。设 MS 是包含下列许可证的多重集： $\{\text{lic}(\text{music}, \Gamma, C), \text{lic}(\text{video}, \Sigma, D)\}$ ，这里：

$$C = \{\text{played_times} \equiv 2\},$$

$$D = \{\text{played_times} \equiv 10\},$$

$$\Gamma = \Sigma = \{\text{canplay}(B, B') : \neg \text{get_value}(B, \text{played_times}, N),$$

$$N < 10,$$

$$\text{set_value}(B, \text{played_times}, N+1, B')\}$$

设 R 是重写规则集：

$$\text{play}(X): \text{lic}(X, \Delta, B) \rightarrow \text{lic}(X, \Delta, B') \Leftarrow \Delta \vdash \text{canplay}(B, B')$$

现在，假设用户要求执行 $\text{play}(\text{music})$ 操作，这样就会匹配规则 $\text{play}(X)$ ，给出匹配 $\sigma_1 = \{X/\text{music}\}$ 。

下一步在 MS 中寻找可能的 $\text{lic}(\text{music}, \Delta, B)$ ，只有唯一的可能匹配 $\text{lic}(\text{music}, \Gamma, C)$ ，这就给出了匹配 $\sigma_2 = \{\Delta/\Gamma, B/C\}$ 。

现在估算条件 $\Delta \vdash \text{canplay}(C, B')$ 。既然 C 中的变量 $\text{played_times} < 10$ ，那么 Prolog 程序 Δ 中的 $\text{canplay}(C, B')$ 成功，条件满足。得到计算的替换 $\sigma_3 = \{B/\text{played_times} \equiv 3\}$ 。

最后，更新 MS，许可证 $\text{lic}(\text{music}, \Gamma, C)$ 从 MS 中移去，用 $\text{lic}(\text{music}, \Gamma, C')$ 替代，这里 $C' = \{\text{played_times} \equiv 3\}$ 。

假设现在请求执行 $\text{play}(\text{video})$ 这个操作，即使在多重集中有一个匹配规则和一个匹配的许可证，也不能执行。这是因为，在许可证 $\text{lic}(\text{video}, \Sigma, D)$ 中的唯一匹配条件 $\Delta \vdash \text{canplay}(B, B')$ 不成立。

下面给出执行步骤的形式化定义。

定义 6-3 执行步骤

设 ms 和 ms' 是两个许可证的多重集， R 是规则的集合，并且假定 a 是一个基项（操作）。记 $\text{ms} \xrightarrow{a}_R \text{ms}'$ 当且仅当存在一条规则 $l: l_{\text{ms}} \rightarrow r_{\text{ms}} \Leftarrow \text{cond} \in R$ 。替换如下：

① a 和 l 匹配，其匹配替换为 σ_1 。

② 对 ms 的某个子多重集 i_{ms} ， $l_{\text{ms}}\sigma_1$ 和 i_{ms} 匹配，其匹配替换为 σ_2 （因此，有某个多重集 rest_{ms} ，使 $\text{ms} = i_{\text{ms}} \cup \text{rest}_{\text{ms}}$ ，并且 $l_{\text{ms}}\sigma_1\sigma_2 = i_{\text{ms}}$ ）。

③ 在第②步之后,对某些程序 P_1, \dots, P_n 和查询 Q_1, \dots, Q_n , $\text{cond} \sigma_1 \sigma_2$ 具有形式 $P_1 \vdash Q_1, \dots, P_n \vdash Q_n$ 。现在对于 $\text{cond} \sigma_1 \sigma_2$ 中的每个 $P_i \vdash Q_i$, 要求 $Q_i \delta_1 \dots \delta_{i-1}$ 在 P_i 中是成功的, 计算其替代 δ_i 。

④ 从 ms 中移去 i_{ms} , 并把多重集 $r_{ms} \sigma_1 \sigma_2 \delta_1 \dots \delta_n$ 加入, 得到 ms' 。

既然 R 和上下文无关, 对 R 省去严格的表示, 把 $ms \xrightarrow{a}_R ms'$ 简单写成 $ms \xrightarrow{a} ms'$ 。定义 6-3 步骤①表示对执行用户给定的请求操作 a (如 $\text{play}(\text{mus})$) 所选择的规则, 原则上, 可能会有多于一个的规则匹配请求, 如果没有规则匹配这个请求, 则请求操作失败。在选择一个规则后, 在步骤②中, 从 ms 中提取许可证 i_{ms} , 和规则的左部匹配。可能会有 ms 的不同子多重集和规则的右部匹配, 这对应于用户有多个许可证允许执行想要的操作。在理论上, 这是一个不确定的因素, 但在实际中, 系统可以向用户提问他要用哪个许可证。步骤③通过执行相关的查询来检查规则中的条件, 每个查询产生一个新的替换, 并传给下一个查询, 如果某个查询失败, 那么整个过程就失败了。最后, 步骤④生成新的多重集。

图 6-22 显示了具有数据 Content 和绑定集 Bindings 的许可证根据规则在多重集的转换。

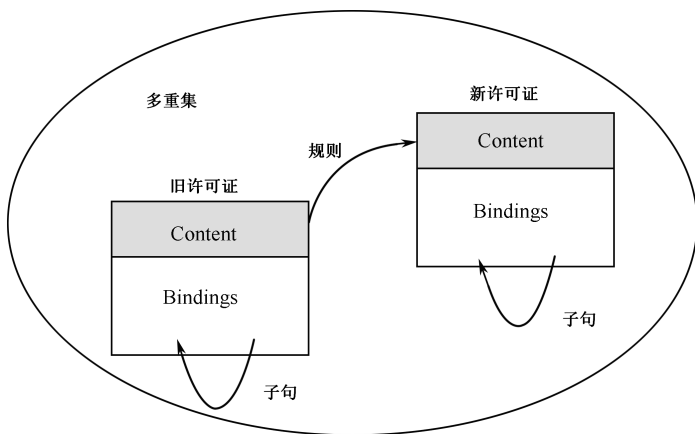


图 6-22 许可证在多重集中的转换

参考文献

- [1] Cheun Ngen Chong, Ricardo Corin, Sandro Etalle, et al. LicenseScript: A Novel Digital Rights Language and its Semantics. Proc of the 3rd International Conference on Web Delivering of Music. Los Alamitos: IEEE ComputerSociety, 2003.
- [2] LicenseScript-A language and framework for calculating licenses on information over constrained domains. <http://es.ewi.utwente.nl/licensescript/>
- [3] XrML Specifications. http://www.xrml.org/get_XrML.asp
- [4] XrML 2.0 Technical Overview, ContentGuard, 2002.
- [5] Open Digital Rights Language(ODRL), <http://odrl.net/1.1/ODRL-11.pdf>
- [6] 陈利颐. XrML 实现第二代数字权益管理机制的研究. 上海交通大学学报, 2003.
- [7] D. Eastlake, J. Reagle. XML Encryption Syntax and Processing W3C Recommendation. <http://www.w3.org/TR/xmlenc-core,2002>.

- [8] D. Eastlake, J. Reagle, D. Solo. XML-Signature Syntax and Processing W3C Recommendation. <http://www.w3.org/TR/xmlsig-core,2002>
- [9] ODRL 权利描述语言逻辑实施机制研究. 计算机科学, 2009, 36(4): 133-139.
- [10] 张芬. XML 加密与签名技术研究及应用, 西安电子科技大学硕士学位论文, 2009.
- [11] 王书锋. XML 文档签名和加密技术研究. 天津大学硕士学位论文, 2003.
- [12] 张辉明. XML 安全的研究与应用. 上海交通大学硕士学位论文, 2007.
- [13] GUNTER C, WEEKS S, WRIGHT A. Models and languages for digital rights. Proc of the 34th Annual Hawaii International Conference on System Sciences. Washington DC: IEEE Computer Society, 2001: 4034-4038.
- [14] R. Pucella, V. Weissman. A logic for reasoning about digital rights. Proc of the 15th Computer Security Foundations Workshop. Washington DC: IEEE Computer Society, 2002: 282-294.
- [15] 韩立龙, 刘清堂, 杨宗凯. 基于逻辑推理的数字权利动态描述研究. 计算机应用研究, 2009, 26(5): 1888-1890.
- [16] 孙伟, 翟玉庆. 一种采用一阶动态逻辑表示的数字权限描述模型. 计算机应用, 2005, 38(4): 846-849.
- [17] 韩立龙, 刘清堂, 杨宗凯. 一种数字权利动态跃迁模型研究. 计算机应用研究, 2009, 26(12): 4740-4743.
- [18] K. R. Apt, D. Pedreschi. Modular termination proofs for logic and pure prolog programs. Proc of Advances in Logic Programming Theory. New York: Oxford University Press, 1995: 183-229.
- [19] 王健宗, 庄超, 蒋文超. 基于 DRM 数字权利描述语言互操作性研究. 计算机与数字工程, 2007, 38(6): 4-6.
- [20] C. N. Chong, R. Corin, J. Doumen, et al. LicenseScript: A Logical Language for Digital Rights Management. Annales des Telecommunications, 2006, 61 (3/4): 284-331.

DRM 应用

目前 DRM 技术的应用领域主要是电子书、流媒体、电子文档等。

从 1999 年开始，出现了以 DRM 为核心的电子书技术，它被评为 1999 年度十大科技成果之一。美国畅销小说作家 Stephen King 在 2000 年 3 月 14 日发表了一本电子书《Riding the Bullet》，这是第一本只出电子书不出印刷版本的书，也得益于 DRM 技术，作家在半月内获得了 45 万美元的收入。DRM 技术是电子书出版中最重要的技术基础，只有通过它，电子书的作者和出版社才能得到相应的收益，DRM 技术推动了电子书产业的发展。

近年来，DRM 技术已较多地应用在音频、视频等流媒体产品的网络传播与销售中。流媒体 DRM 系统既要保证流媒体的实时性，适应复杂多变的网络状况以及服务器的传输控制策略；又要保证流媒体的安全性，涉及数字版权的描述、用户身份的认证和管理、流媒体内容的安全分发等方面^[1]。微软公司的 WMRM (Windows Media Rights Manager) 是市场上完整的 DRM 解决方案，包括实现 DRM 的体系结构和二次开发的 SDK。RealNetworks 公司的 Helix DRM 是一个综合、灵活的平台，确保 RealAudio、RealVideo、MP3、MPEG-4、AAC、Sony 的 ATRAC3 和 H.263 等格式媒体的安全传输。Helix DRM 包括一系列的产品与设备，它使版权所有者能创建一系列健全的商业模式来传输多媒体给世界各地的观众，并为消费者找到和欣赏数字媒体提供了很多新的方式。

在电子文档应用领域则更多体现了权限管理的特点，既要考虑版权，更要注重文件的访问控制，因此这是一种更高层次的数字权限管理^[1]。常见的基于 DRM 技术的电子文档格式主要有两大类型：一种是非固定版式的文件格式。这方面的 DRM 产品，常见的有对 Office 文档的保护和对 HTML 格式的保护，微软的 Office 2003 就带有含 DRM 技术的 IRM 服务，保护对象是 Word、Excel、PowerPoint 文档。另一种电子文档格式是版式文件，PDF 是目前市场上使用较多的版式文件。以 PDF 为格式的电子文档 DRM 产品中，最为著名的当属美国 Authentica 公司的 Secure Documents for PDF 系统。在中文版式方面，北大方正的 CEB 格式是一个比较突出的版式，它在 DRM 技术的基础上进行设计，在 DRM 体系方面有较好的支持。

7.1 流媒体的 DRM

7.1.1 流媒体介绍

流媒体指在网络中使用流式传输技术的连续时基媒体，如音频、视频或多媒体文件。流式传输技术相对于传统下载方式的优点在于采用这种方式时，用户不必等到整个文件全部下载完毕，而只需经过几秒或几十秒的启动延时即可进行播放和观看。此时多媒体文件的剩余部分将在后台从服务器继续下载。

影响在线音频、视频等传输实时性的主要是频带宽度限制、传输品质不稳。通常，网络服务商（ISP，Internet Service Provider）所提供的频宽只是理论最大值。实际上，网络固有的互连而不互管特性导致用户使用频宽并不固定，即使正常情况下也会有 10%~30% 的差异。这样，必然影响媒体播放的实时性。

为解决信息传输实时性问题，开发了流式传输及流媒体。前者主要指通过网络传输媒体（如音频、视频等）的技术总称，其特定含义为通过网络将音频、视频等传输到用户终端播放时，无须等全部文件下载完毕才可播放，而是将连续的音频、视频信息压缩后放于网站服务器，用户终端播放时只要将开始部分的内容存入其内存，其余数据流由用户终端在后台继续接收并播放，直至播放完毕或用户中止操作。这样，只在开始时有几秒延迟，而用户播放媒体的等待时间将显著减少，且不需要太大缓存。后者指使用流式传输技术的连续时基媒体，如音频、视频、多媒体文件，其实现的关键技术是流式传输。

采用流媒体技术，无须将全部信息下载再播放，因而等待时间大为缩短；流文件小于原始文件数据量，且用户无须将全部流文件下载到硬盘，从而节省了大量磁盘空间；采用实时传输协议，更适合音频、视频等网络实时传输。正因为上述显著优点，流媒体技术已受到业界普遍关注，著名的 IBM、Intel、Microsoft、康柏等都在调整其战略方向，以便在该市场占据有利地位。另外，从用户角度看，超过三分之一的网络用户曾阅读过流式内容，显示对丰富媒体的需求必将推动整体架构的革新^[2]。

在业界，有多家公司提供音视频 DRM 解决方案：微软公司的 WMRM（Windows Media Rights Manager）和 RealNetworks 公司的 Helix DRM，以及 InterTrust 公司的 Rights System 和 IBM 公司的 EMMS（Electronic Media Management System）等。

7.1.2 WMRM

1. WMRM 的基本工作流程

微软发布了数字版权管理的开发包——WDRM SDK。WMRM 是微软为 Windows 流媒体平台开发的基于 XrML 端到端的 DRM 系统。它支持大量的安全特性和商业使用模式选项，并且使用 WMRM 进行开发和分发是免费的。目前所受的限制是 WMRM 只支持微软自身的 WMA、WMV 格式的音视频文件，并且需要为许可证服务器从微软公司申请一个许可（license）并每年更新一次。

微软针对数字音视频服务的 DRM 方案主要包括服务器端的 WMRM 和客户端支持 WMRM 的 Media Player。WMRM 开发包提供了服务器端和客户端的 SDK（软件开发包），它使用了 COM 对象技术，允许开发者开发加密的媒体文件和发布证书的应用程序。这个特性

能够将 WMRM 和已有的商业应用程序相结合。

一个相关的技术是 Windows Media Device Manager, 它允许用户将受保护的文件传送到便携式设备（如硬件媒体播放器）中。利用 Windows Media Player, 我们可以将受保护的 Windows 媒体文件传送到大多数的兼容 SDMI (High Definition Multimedia, 高清晰度多媒体接口) 的硬件设备中。

图 7-1 是 WMRM 的工作流程图^[3]。

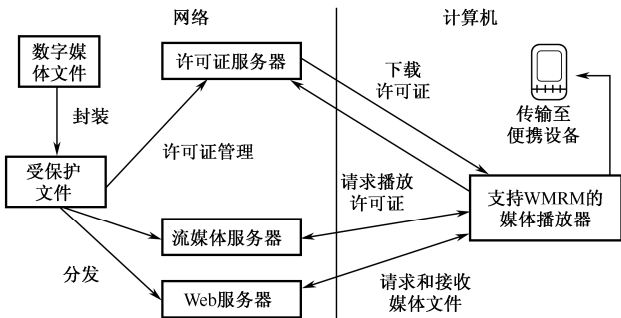


图 7-1 WMRM 的工作流程图

其主要步骤如下。

（1）封装

WMRM 通过对媒体文件加密和设置特定文件头来封装数字媒体文件。封装的文件将加密并使用一个“密钥”锁定。该密钥存储在一个加密许可证中，该许可证将单独分发。它还会向数字媒体文件中添加其他信息，例如用于获取许可证的 URL。打包的数字媒体文件将保存为 Windows Media Audio 格式（文件扩展名为.wma）或 Windows Media Video 格式（文件扩展名为.wmv）。

（2）分发

封装的文件以多种方式发送，如置于一个网站以供下载，放在一个媒体服务器以提供流媒体服务，甚至以光盘、E-mail 方式发送等。

（3）建立许可证服务器

内容提供商可设置许可证条款，建立一个许可证服务器，存储许可证的特定权限或规则，并提供 Windows Media 权限管理器许可证服务。客户端请求和接收媒体文件，用户在播放受 WMRM 保护的 Windows 媒体文件时需要向服务器发出许可证请求，许可证服务器对请求许可证的消费者进行身份验证。数字媒体文件和许可证是分开存储和分发的，因此更便于管理整个系统。

（4）获取许可证

要播放封装的数字媒体文件，用户首先必须获取一个许可证密钥为该文件解锁。当用户试图获取封装的数字媒体文件、获取一个预先传递的许可证或首次播放该数字媒体文件时，都将自动启动获取许可证的过程。在满足相关条件的情况下，服务器可以显式或隐式地分发播放许可证，或引导用户进入注册页（该页要求输入信息或付费），或从许可证服务器检索一个许可证而不提示任何问题。

（5）播放数字媒体文件

要播放数字媒体文件，用户需要一个支持 WMRM 的播放器。用户可根据许可证中所提

供的规则或权限来播放媒体文件，在许可证许可的情况下，授权用户可以将音频资料传输至支持 WMRM 技术的便携设备上播放。许可证可提供多种不同权限，如开始时间和日期、持续时间以及对操作计数。例如，默认权限可能允许消费者在特定计算机上播放数字媒体文件并可将该文件复制到便携设备上。但是，许可证是不可转让的。如果便携设备的音频资料再次传输到其他计算机上，则必须重新获取许可证才可以播放。这种按 PC 颁发许可证的模式可确保打包的数字媒体文件只能在已获得该文件的许可证密钥的计算机上播放。

2. WMRM 的系统结构

基于流媒体的播放流程和可实现的功能，WMRM 系统分为以下三个部分：视频文件打包器、证书发放器和数据库三部分（图 7-2）^[2]。

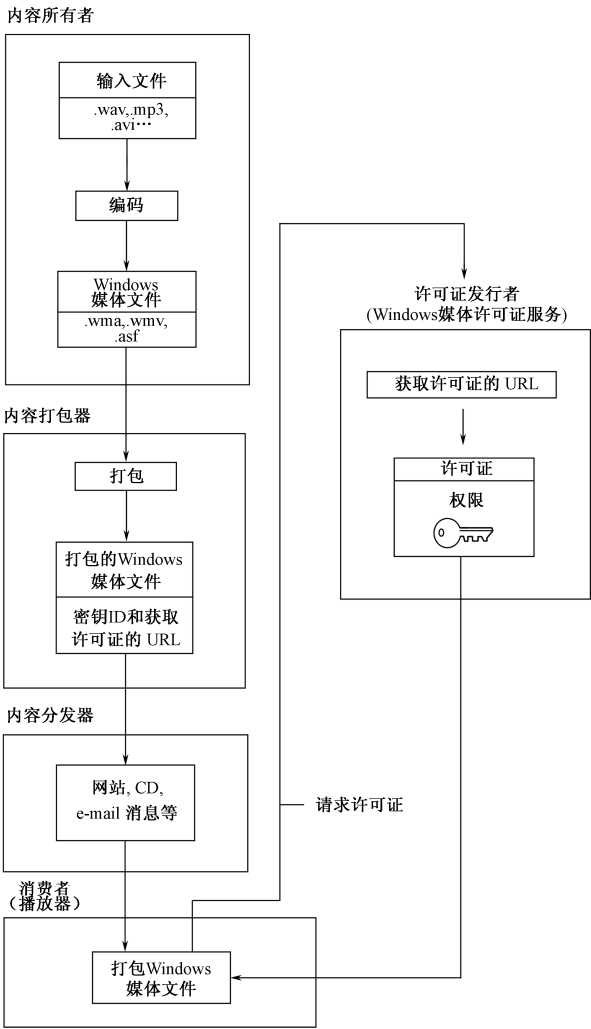


图 7-2 WMRM 系统结构

视频文件打包器对视频文件进行封装，将未加密的视频文件，经过加密使得其需要证书方可播放。这样来保证只有授权用户有播放权限，以达到版权保护的目的。

证书发放器即许可证服务器，用户向许可证服务器发送申请证书的请求，许可证服务器

发放制定流媒体文件的证书。出于安全考虑，为了防止伪造请求的发生，一般都会在此系统中添加时间戳的校对。

数据库是 DRM 系统最重要的部分，存储了所有流媒体文件的加密信息，如需要的话，还可记录用户信息、点播状况信息等。当打包器加密文件的同时，将所有加密数据信息输入数据库中，而在用户请求证书时，证书发放中心会依据媒体文件序号从数据库中查找相应加密信息，以生成证书并发放给用户。

在微软的 WMRM 方案中使用了基于对称和不对称算法的加密机制进行内容及其相关许可证的加密，使用计算机的硬件标识信息对客户端播放软件进行个体化，使用水印技术来跟踪媒体内容的使用，使用自我保护容器技术来防止对于保护数字内容的非授权访问。针对权限可移植性的问题，微软的 DRM 方案使用了混合用户和设备的标识方案，即将用户的唯一标识与使用该用户标识激活的不同设备上多个客户端软件实例关联，以模拟一个用户拥有多个设备的情况。

目前微软的 WMRM 系统遵循 SDMI (Secure Digital Music Initiative, 面向计算机和各种数字设备上数字音频的开放的权利保护规范和技术框架) 标准，选择 XrML 为权限元数据方案，采用了私有的内容标识体系。

3. WDRM SDK 权限管理对象

WMRM SDK 是微软发布的数字版权保护的开发包，图 7-3 大致描述了 WMRMHeader、WMRMProtect 和 WMRMKeys 这些权限管理对象在打包加密中的作用^[2]。

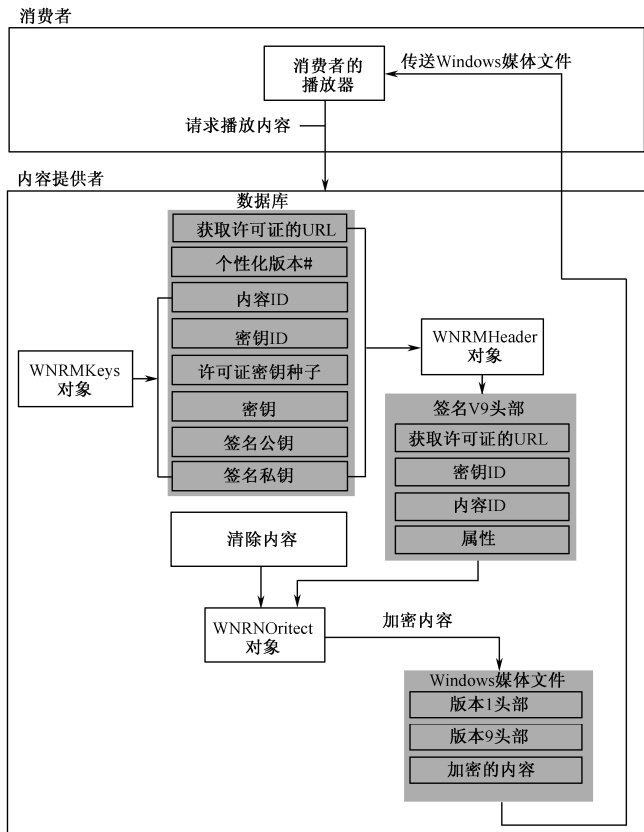


图 7-3 WMRM各个对象使用示意图

其他权限管理对象在打包、取证的过程中也十分重要，由于开发包中所用到的 WMRMChallenge、WMRMHeader、WMRMLicGen、WMRMRights、WMRMKeys、WMRMResponse、WMRMProtect 等对象在整个系统中起到了很重要的作用，因此对该开发包中权限管理对象进行大致介绍。

(1) WMRMChallenge 对象

WMRMChallenge 对象存储一个客户的请求信息，包括：

创建 Key 使用的 KeyID；

鉴定播放内容的 ContentID；

确定播放内容的属性；

获取验证的 URL 地址；

播放内容的 DRM 版本；

加密过的客户 ID (ClientID)；

客户机的 DRM 版本；

客户机的个性版本号。

该对象属性如表 7-1 所示^[2]。

表 7-1 WMRMChallenge 对象属性表

属 性	注 释
Action	获得 Challenge 请求的指定 action
ActionCount	获得 Challenge 请求的 action 数量
Challenge	指定和获得证明书请求
ClientAttribute	从客户信息部分中获得指定名、值对
ClientInfo	获取 Challenge 的客户信息部分
Header	获得 Challenge 的头部信息
V1Challenge	从 7 版本 Challenge 中获得 1 版本证书请求

(2) WMRMHeader 对象

WMRMHeader 对象包含打包内容的头部信息，如果想要创建多媒体内容为 Windows Media 的格式，那么必须为这个文件创建一个头部信息，它包含以下内容：

创建 Key 使用的 KeyID；

鉴定内容使用的内容 ID；

内容属性；

获得认证信息的 URL 地址；

应用程序安全信息。

该对象属性如表 7-2 所示^[2]。

表 7-2 WMRMHeader 对象属性表

属 性	注 释
Attribute	获取和指定名、值对
ContentID	获取和指定加密内容的 ID
Header	获取和指定内容头部分
IndividualizedVersion	获取和指定 WMDRM 最小版本
KeyID	获取和指定加密密钥的 ID
LicenseAcqURL	指定和获得证书 URL 地址
Version	获得版本号

(3) WMRMKeys 对象

WMRMKeys 对象用来生成头部信息必须的内容。

该对象属性如表 7-3 所示^[2]。

表 7-3 WMRMKeys 对象属性表

属 性	注 释
KeyID	获得和指定 KeyID
Seed	获取和指定证书密钥种子

(4) WMRMLicGen 对象

验证机构需要使用 WMRMUcGen 对象来创建一个播放许可证，一个播放许可证包含以下信息：

- 加密保护内容的 Key 值；
- 赋予请求客户的播放权限；
- 客户机的信息；
- 不是必需的名称信息；
- 客户播放的许可证优先权；
- 许可证信息。

该对象属性如表 7-4 所示^[2]。

表 7-4 WMRMLicGen 属性表

属 性	注 释
Attribute	获取和指定名、值对
BindToPubKey	获取和指定公钥，并绑定证书
ClientInfo	获取和指定客户计算机信息
KeyID	获取和指定加密密钥的 ID
Priority	获取和指定优先级
Rights	获得版权信息

(5) WMRMProtect 对象

WMRMProtect 对象负责加密内容，并将加密内容与头部信息打包至一个 Windows Media 文件。

该对象属性如表 7-5 所示^[2]。

表 7-5 WMRMProtect 对象属性表

属 性	注 释
Header	获取和指定内容头部分
InputFile	获取和指定输入文件名
Key	获取和指定内容加密密钥
V1KeyID	获取和指定 1 版本 Key ID
V1LicenseacqURL	获取和指定 1 版本证书获取地址

(6) WMRMResponse 对象

WMRMResponse 对象负责创建一个认证的回复。

该对象属性如表 7-6 所示^[2]。

表 7-6 WMRMResponse 对象属性表

属 性	注 释
ReplaceQuotesWith	指定由 GetLicenseResponse 方法获得的数学串的替代

该对象方法分别如表 7-7 所示^[2]。

表 7-7 WMRMResponse 对象方法表

属 性	注 释
AddLicense	向证书响应添加证书
GetLicenseResponse	获取包含证书的证书响应

其中 AddLicense 需要指定 WDRM 版本参数，该参数如表 7-8 所示^[2]。

表 7-8 AddLicense 版本参数表

值	含 义
1.0.0.0	指定 WDRM 的 1 版本
2.0.0.0	指定 WDRM 的 7 版本

(7) WMRMRights 对象

WMRMRights 对象指定客户拥有的权限，如播放次数、播放环境等。该对象属性如表 7-9 所示^[2]。

表 7-9 WMRMRights 对象属性表

属 性	注 释
AllowBackupRestore	指定和获取布尔值，表示证书是否允许存储备份
AllowBumToCD	指定和获取布尔值，表示证书是否允许内容以 RedBook Audio 格式刻成 CD

续表

属 性	注 释
AllowPlayOnPC	指定和获取布尔值，表示证书是否允许内容在客户 PC 上播放
AllowTransferToNonSDMI	指定和获取布尔值，表示证书允许内容复制到 SDMI 不兼容的移动设备或移动媒体
AllowTransferToSDMI	指定和获取布尔值，表示证书允许内容复制到 SDMI 兼容的移动设备或移动媒体
BeginDate	指定和获取证书最早有效时间
BurnToCDCCount	指定和获取内容可以刻成 CD 的次数
DeleteOnClockRollback	指定和获取布尔值，表示当时钟被调回时证书是否必须被删除
DisableOnClockRollbach	指定和获取布尔值，表示当时钟被调回时证书是否必须无效
ExpirationDate	指定和获取过期日期
MinimumAppSecurity	指定和获取最小安全级别
Play Count	指定和获取内容可播放次数
PMAppSecurity	指定和获取已传送到移动设备或移动媒体的内容的安全级别
PMExpirationDate	指定和获取媒体证书的过期时间
PMRights	指定和获取移动证书管理的权限
TransferCount	指定和获取内容可转移到移动设备或移动媒体

(8) LicenseGenerator 对象

LicenseGenerator 对象用来生成 WMRM 1 的许可证，并不适用于 WMRM 7。
该对象属性如表 7-10 所示^[2]。

表 7-10 LicenseGenerator 对象属性表

属 性	注 释
DeleteIssueEnty	删除 1 版本证书数据库条目
Get	获取证书属性值
InstallKeys	安装新的公钥私钥到数据库
IssueLicense	通过 challenge 创建证书
Set	指定证书属性

7.1.3 Helix DRM 方案

Helix DRM 是 Real Networks 公司推出的针对数字音视频服务的 DRM 解决方案，支持的格式有 Real Video、Real Audio、MP3、AAC、MPEG-4、H.263、AMR，它可以将以上内容发送到传统 PC、数字机顶盒、手机、个人信息终端和便携式音乐播放器中。Helix DRM 包含了一系列的产品和服务手段，通过安全加密手段来保护版权，将电影和音乐发售到全世界的 Internet 用户手中。

1. Helix DRM 系统结构

Helix DRM 能够实现的功能有：媒体内容加密封装，生成授权证书，高质量的媒体内容传输至授权的播放器中。目前 Real 播放器已经具有多个主要的平台的版本，如 Windows、Linux、UNIX 等，在 PDA、手机中也有 Real 播放器。

Helix DRM 基于一种模块化、开放、可扩展、可伸缩的架构，具有容易与现有系统集成、支持不同实施场景和多种媒体文件格式、对 Windows 和 UNIX 的跨平台兼容等特点，主要包括服务器端的 Helix DRM 打包、内容服务器、许可证服务器和客户端 Real One Player 播放器的安全文件格式插件等（图 7-4）^[4]。

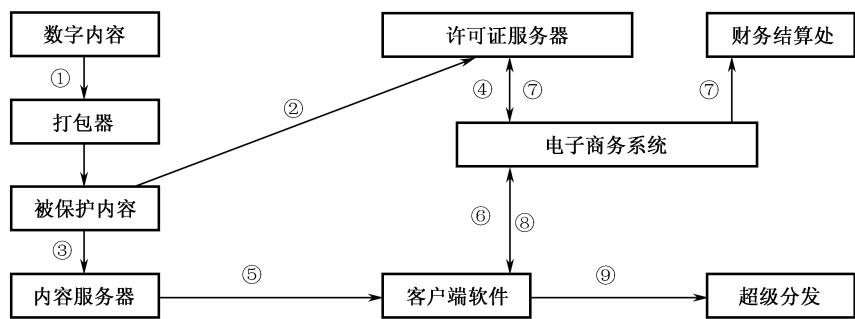


图 7-4 Helix DRM 系统结构图

（1）Helix DRM 打包器

在 Helix DRM 方案中，封装通过 Helix DRM Packager 完成。Helix DRM Packager 使用强大的加密算法和安全组件容器技术，防止用户非法使用内容，为内容通过流式传输、下载或其他传输方法分发做准备。打包器加密内容文件，而打包的媒体内容和与之关联的打开、使用内容的商业规则是分开存储的，因为权限和内容在产生一个许可证前不耦合，内容提供商能根据需要动态规定媒体内容的使用权限。Helix DRM 打包器支持多种媒体格式，与 Helix Producer 一起使用，可以安全传输直播内容。

（2）Helix DRM 许可证服务器

Helix DRM 许可证服务器是一个可扩展、灵活的服务器，允许零售商、Internet 音乐和电影服务商和企业管理，发放许可和报告内容交易。Helix DRM 许可证服务器以一种简单 HTTP 格式，接收和验证对于许可证的请求，产生允许访问受保护媒体的许可证，它能够针对相同内容提供不同类型的许可证给不同用户。这方便了零售商网络服务器和 Helix DRM 许可证服务器的集成和通信。

Helix DRM 许可证服务器发放的许可证具有如下特点：与一个客户端设备唯一绑定，在发送中被认证和保证安全，支持权限的灵活分配，能够用来撤销或者恢复许可证（如果内容提供商选择实施这些能力）。每个许可证对于请求的可信任用户所在的设备是唯一的，以防止传输许可证给其他用户。另外，内容钥匙总被加密。

（3）内容提供者

安全的 Real 媒体文件几乎能够通过任何分发机制分发：FTP 下载、P2P 网络、组播甚至 CD 等。这是权限与内容文件分开存储的另一个原因。不管内容怎样发送，它仅仅能够使用发给单个客户端的一个有效许可证播放。因为经过安全处理的内容文件与内容钥匙的存储分开，在文件所在位置是不能观看内容文件的。一旦用户获得一个许可证，他接收内容；同时，带有安全插件的 Real One Player 在他的许可证中检索相关的安全钥匙来播放内容。

(4) 电子商务系统

由零售商网络服务器（一种现有的前端网站）提供，消费者通过它请求到安全内容的许可证。零售商网络服务器转发这些请求到 Helix DRM 许可证服务器，并且返回 Helix DRM 许可证服务器产生的许可证给消费者。零售商网络服务器需要发送客户端特定信息，从内容数据库检索到的内容唯一标识符和内容钥匙，并且添加针对这个用户和内容的适当权限。零售商网络服务器能够使用任何方法来从可信任的客户和内容数据库检索这种信息。例如，零售商网络服务器能够嵌入播放器功能或者使用一些脚本语言检索客户信息，使用与内容数据库格式兼容的任何数据库操作语言从内容数据库中检索信息。零售商网络服务器仅仅需要在发送这种信息到 Helix DRM 许可证服务器时遵循 HTTP 请求格式。

零售商网络服务器作为可信任客户和 Helix DRM 许可证服务器之间的一个代理，被授权接收信任客户的许可证请求，并向许可证服务器请求许可证。采用这种方式能够防止消费者与 Helix DRM 许可证服务器之间的直接交互，有助于避免对 Helix DRM 许可证服务器的无效请求或者篡改 Helix DRM 许可证服务器的尝试。

(5) Helix DRM 客户端

Helix DRM 客户端可以在反篡改环境里下载和播放安全格式内容。这个环境是基于内容拥有者定义的使用规则的。客户端程序（如 Real One Player）可根据 Helix DRM 客户端标准开发制作。

客户端是带有安全插件的 Real One Player，一种可信任客户端，能够识别安全的 Real 媒体文件。安全的 Real 媒体文件是 Real 媒体文件转换为以.rms 格式存储的安全数据包。rms 格式是专门为安全媒体开发的。带有安全插件的 Real One Player 执行客户端媒体引擎保证了整个系统完整性，并且确保内容仅仅能够在一种可信任的、防篡改的环境中播放。这个可信任客户具有一个安全的权限数据库，在其中存储检验和跟踪客户端目前所有权限需要的所有许可证。用户以在线或者离线方式播放媒体，客户端总是能够执行与权限联系的商业规则和相关权限。如果一个用户没有有效的许可证而试图播放内容，播放将被禁止，用户被送往零售商网络服务器以获取一个有效许可证。信息仅在用户和内容的许可证发放方之间发送，以保护隐私。

(6) Helix Universal Server DRM 插件

这个插件能使 Helix 通用服务器中受保护的媒体进行流式传输。

2. Helix DRM 的工作流程

① Helix DRM 打包器将一种不安全的 Real 媒体文件转换为一种安全的媒体文件。它同时产生内容文件的一个全球唯一标识符（GUID）和一把安全钥匙，并将它们保存在一个文本文件中，以导入零售商的内容数据库。

② 内容 GUID 和安全密钥导入一个后端数据库以建立许可证服务器。

③ 通过一些发送机制，如一个媒体服务器、CD，分发安全媒体文件到消费者手中。

④ 零售商设置许可内容的使用规则。

⑤ 用户通过上述发送机制检索媒体文件。

⑥ 用户联系零售商网络服务器，以获取播放安全媒体文件的一个许可证。

⑦ 零售商网络服务器从许可证服务器请求权限，一个结算中心处理用户获取这些权限的财务交易。

⑧ 零售商网络服务器返回许可证给可信任客户端。客户端检查它的安全许可证数据库，以确保它已经接收到播放文件的权限。仅在那时可信任客户端将播放文件。

⑨ 安全内容文件可以超级分发，但是播放需要授权，步骤同⑥～⑧。

3. Helix DRM 的安全机制

在 Helix DRM 方案中使用了强加密算法进行内容及其相关许可证的加密，使用水印技术来跟踪内容的使用，使用自我保护容器技术来防止对于保护数字内容的非授权访问，同时使用客户端个体化和防篡改机制加强安全。针对权限可移植性的问题，Helix DRM 允许用户备份他们的许可证，并且恢复到另外一台设备。为了防止滥用，用户仅仅能够进行固定次数的这种操作。

Helix DRM 遵循 SDMI 标准。目前 Real Networks 采用了私有的内容标识体系和权限元数据方案，以后可能选择 ODRL。

7.2 电子书的 DRM

电子书的 DRM 技术相对比较成熟，国内外的应用也较多。国外的电子书 DRM 系统有 Microsoft DAS、Adobe Content Server（原 Glassbook Content Server）等，国内的电子书 DRM 系统有方正 Apabi 数字版权保护系统^[7]、书生的 SEP 技术、超星的 PEG 等。

7.2.1 电子书的发展概况

随着 Internet 的迅速普及与信息存储、检索技术的不断完善，在实现书目、文摘、索引等二次文献的数字化的基础上，电子出版商和文献机构致力于期刊、专利、会议录、学位论文等一次文献的全文数字化产品及其服务系统的开发，网络已经成为人们获取这些文献类型的重要途径。早期的电子书通过光盘或 Internet 免费下载等形式提供给读者，版权无法得到保护，因此电子书出版发展非常缓慢，光盘出版营利艰难。DRM 概念的提出，为在电子书出版、发行、销售、使用过程中各方权利的实现与保护提供了可行的技术路线。电子书阅读器与 DRM 技术的结合，为有效解决电子书阅读障碍提供了技术保障。

从 2000 年《Riding the Bullet》发表以来，电子书保持着持续的发展。据统计，美国 80% 以上的出版社都进入了电子书出版领域。在 2001 年，只有方正电子和美国的 Net Library，推广基于 DRM 技术的数字图书系统，把有版权保护的电子书应用到图书馆。而从 2002 年年底开始，更多的技术及服务提供商推出了针对电子书的产品，例如 Adobe、Microsoft 等。

针对电子书的 DRM 技术，自 2000 年以来逐渐普及。国内外的电子书 DRM 系统虽然在实现方法上有所区别，但都具备一些共同的特点，例如对电子书的复本数进行控制，使电子书像纸书一样按“本”销售，读者可以购买电子书或者借阅电子书等。电子书的 DRM 技术虽然比软件的保护技术出现得晚，但电子书内容本身只是数据，不像软件的代码可以被跟踪

执行,所以对电子书的版权保护比软件的保护更有效。DRM 技术保护是电子书出版中最重要的技术基础,只有通过它,电子书的作者和出版社才能得到相应的收益,电子书销售数量才可计数。电子书销售网站能从出版社得到电子书的销售许可,读者通过网上支付购买电子书。出版社也可以把电子书通过销售渠道卖给图书馆,图书馆购买电子书就像购买纸书一样,按复本数购买,对读者提供借阅的服务。DRM 技术使电子书与普通的电子文档相比,具有了复制代价更高和以“本”进行销售、统计的特性,更像我们所熟悉的纸书了,从而能够保证电子书行业的可持续发展^[1]。

由多家著名电子出版商、电子产品制造商、网络书店及美国图书馆协会等组成的“电子书交换工作组”(EBX Working Group),为保护电子书版权,制定了《EBX 系统标准》,2000年7月发布了其最新的0.8版标准草案^[5]。该标准详细说明了电子书从出版、发行、销售到使用的全过程中版权保护的技术方法,规定了有关各方的职能和权益。为了对电子书的版权进行有效管理,EBX 提出通过创建与转移数字化“凭证”(Voucher)的方法来实现。EBX 所谓的“凭证”,是指许可电子书在网络中从一个持有者转移到另一个持有者的数字化对象。电子书的“凭证”用 XML 编码,它包括以下信息:

- ① 标识(ID):电子书的 ISBN(国际标准书号)或 DOI(数字对象标识符)。
- ② 内容密钥:数字化内容解密密钥(如 56 位 DES),用于对加密的电子书内容文件解密。
- ③ 复制计数:允许数字化凭证持有者浏览、外借、赠送或出售的电子书内容的拷贝数,对于数字图书馆来说,相当于采集的电子书的复本数。
- ④ 许可:数字化凭证持有者对电子书具有不同的处置许可,包括如下内容。
 - 外借:规定数字化凭证持有者是否许可将凭证外借他人,允许他人在一定期限内获得阅读该电子书内容的权利。
 - 赠送:规定数字化凭证持有者是否许可将凭证赠送,使受赠者成为凭证的持有者。
 - 出售:规定数字化凭证持有者是否许可将凭证出售。
 - 外借期限:规定数字化凭证持有者允许外借凭证的最长时间。
 - 个人使用时间:规定数字化凭证持有者使用电子书的时间,以天、周、月或年计。
 - 个人使用复制数:规定在个人使用时间内对电子书内容所许可复制的最大数量。
 - 个人使用复制大小:规定个人使用电子书时许可复制的内容大小,即许可复制段落、整页、整章或全部内容。

EBX 提出的 DRM 技术的核心是,将电子书阅读器与实施版权保护及使用许可的数字化凭证相结合构成电子书阅读系统,每个电子书阅读系统在制造时都有特定的成对公钥和私钥。电子书内容服务器在确认用户的公钥及其相应的购买或使用权利的有效性后,使用公钥对电子书内容密钥加密,将加密的电子书内容及其凭证传输给电子书阅读系统,已加密的内容密钥只能被该电子书阅读系统对应的私钥解密。当用户阅读电子书时,私钥即对电子书内容密钥解密,内容密钥则对电子书内容文件解密,从而在电子书阅读器中显示出与纸质图书类似的页面。上述过程由电子书服务器和用户的阅读器自动快速完成。电子书阅读系统的私钥即使对数字化凭证的持有者也是保密的,以确保电子书内容在传播、使用过程中的版权保护。因此,对于一种电子书内容,只有经确认授权的阅读系统才能阅读,从而遏制了盗版电子书的传播,确保了合法用户的正常使用。

电子书阅读器的日益普及, DRM 技术的逐步成熟, 将从根本上解决制约电子书发展中的阅读不方便和版权保护困难这两个基本问题。国际著名的大型出版商如 Simon & Schuster、Random House、Harper Collins、St. Martin's Press、McGrawHill、MacMillan Publishing 等都涉足电子书出版, 并与电子书阅读器制造商达成了内容发布与版权保护的协议。

目前, 电子书的出版标准包括 PDF、HTML、XML、OEB 以及电子书阅读器专用的一些格式, 不同制造商生产的电子书阅读器往往与其特定的内容相匹配, 它们之间的互操作性较低。电子书的进一步发展有赖于出版标准的统一和阅读器的兼容^[6]。

7.2.2 Microsoft 电子书系统

1. Microsoft Reader 阅读器

2000 年, 微软推出电子阅读器 Microsoft Reader, 用于在液晶显示屏上进行电子书阅读。当时, Adobe Acrobat 打败了其他竞争产品, 渐渐成为电子书市场实质上的标准格式; 同时, 一批新的竞争对手也开始出现, 推出了其他的 PDF 相关软件, 包括 Ghostscript、Foxit 与 Nitro PDF 等。Microsoft Reader 是 Adobe Acrobat 众多对手中实力较强的一个, 它的文件采用 LIT 格式。

Microsoft Reader 推出后, 成为了微软大力推广的读书软件, 用户可以在微软的网上书店购买电子书, 然后利用 Microsoft Reader 来阅读, 或者将电子书下载到掌上型电脑上, 利用掌上型电脑的 Microsoft Reader 软件阅读。

为了避免自己的电子书重蹈 MP3 音乐的覆辙, Microsoft 公司在 Reader 上实现初步的数字产权管理功能, 此外他们还推出了自己的数字产权服务器 (Digital Asset Server, DAS), 为今后更全面地支持数字产权管理打下了基础。

DAS 是基于数字版权管理的电子书发行和销售管理的企业级产品, 企业用户可以用它提供的功能实现通过网络接受消费者对电子图书的购买要求, 对图书进行加密并跟踪已售出图书的传送过程。

在最初的时候, 依靠 Microsoft 的品牌影响力, Microsoft Reader 与 Adobe Acrobat 势均力敌。但在随后的发展中, 市场对 Microsoft Reader 越来越冷漠, 其 LIT 格式文件也未能受到用户的欢迎。

Microsoft Reader 主要面向传统 PC 和采用 Windows 系统的平板电脑。2005 年, 为了吸引更多用户, Microsoft Reader 推出兼容 Windows Mobile 6.1 的新版本, 但 Microsoft Reader 依然未能得到广大用户的认同, 并渐渐淡出人们的视线。2011 年 8 月, Microsoft 宣布一年后将不再提供 Microsoft Reader 的下载服务, 其对应的 LIT 格式文件也会在应用商店中下架。

对 Microsoft Reader 来说, 这样的发展轨迹似乎并不难理解。用户不喜欢它的很重要的原因是, 很多 LIT 电子书都仅存于没落的电子书应用商店, 例如 ebooks.com 和 Fictionwise 等; 另外, LIT 电子书大部分都不是畅销书籍。

2. Microsoft 电子书系统构成

(1) 数字产权服务器 (DAS)

DAS 包括两个基本的组成部分: DAS e-commerce 和 DAS Server。DAS e-commerce 安装

在网络书店中，用来接收并确认用户的下载请求，DAS Server 是 DAS 的核心，安装在出版社或发行商的服务器上，通过分析 DAS e-commerce 反馈的信息，确认合法的下载请求，并保证已购买的电子书可以安全地下载给用户。

（2）阅读器（Reader）

Reader 作为数字权管理的用户端，下载后必须经过一个激活过程（Activation）才能使用。所谓激活，实际上是将读者的信息登记到 Microsoft 巨大的用户库中。

Microsoft 电子书系统的特点如下。

① 一个 Microsoft Passport 最多能在两台不同的计算机上激活 Reader。但如果一个读者有两个 Passport 账号 A 和 B，他用账号 A 购买的电子书，将不能在账号 B 激活的 Reader 上阅读。难怪有人在讽刺这一管理策略的时候说，买一本 Microsoft 电子书作为礼物送给朋友是不可能了。

② Microsoft 公司将 XrML 视为其数字权管理战略的基础，并且计划在今后的产品中提供更全面的支持。

③ Microsoft 公司定义了 3 种安全层次，分别是：密封（Sealed），对电子书加密，使电子书的内容不被修改；戳记（Inscribed），在电子书首页突出显示获取途径及购买者的信息；所有者独占（Owner Exclusive），只有合法的购买者才能阅读。

7.2.3 Adobe 电子书系统

PDF 是与 Word、HTML 一样广泛流行的电子文档格式，且更加适合较专业的出版印刷应用。基于 PDF 格式的电子书是 Adobe 公司的重要产品之一，借助其 DRM 功能可以实现电子图书加密，按照顾客身份的不同提供不同程度的复制及打印权限，以保护出版商的版权。

Adobe 电子书系统构成如下。

1. Adobe 的 DRM 软件

Adobe 的 DRM 软件包括 Adobe PDF Merchant 和 Web Buy。使用带有 Web Buy 的 Adobe Acrobat 和 Acrobat Reader 的顾客可以按出版商选择的分发方法访问已下载的 PDF 文件。当使用者购买一本电子书时，Web Buy 会用来确认购买者的购买行为，并将购买者标识为具有访问权限的用户。Adobe PDF Merchant 是基于服务器的软件，它充分利用 XML 语言，可以直接集成到电子商务和交易服务器中，用来为 PDF 文件指派访问权限和加密钥匙。PDF Merchant 会在传送的 PDF 文件里加上一把“锁”。当点击购买的按钮时，PDF Merchant 会带你到出版商的网站上，请你输入信用卡的相关信息以购买该书。

如果你想把买过的书籍传送给别人，他们也无法开启。PDF Merchant 会提醒使用者没有阅读的权利，必须上网购买授权。

2. Adobe 内容服务器

Adobe 内容服务器可以实现电子图书交易自动化并且确保电子书销售过程的安全。它可以直接从出版商或销售商的 Web 站点对 Adobe PDF 格式的电子书进行包装、保护、发行和销售。

3. Adobe Acrobat 电子书阅读器

Adobe Acrobat Reader 为通用的 PDF 浏览软件，而 Adobe Acrobat 电子书阅读器则是专为阅读电子图书而设计的，它包括 Read、Library、Bookstore 三个部分，分别为用户提供阅读、建立个人电子图书馆、在线购买电子图书的功能。

Adobe 电子书系统的特点如下。

① 只允许在购买时执行下载的设备上阅读，但用户也可以借出或者转让电子书。由于该技术采用了不能同时多台设备上阅读的机制，因此，在将电子书借出期间，借出方将不能阅读，只有借入方才可以阅读。

② 可以限制电子图书的打印、指定阅读时间范围等。

③ 具有的 B2B 特性支持图书出版商、零售商或者分销商之间的图书交易，利用加密的电子书，Adobe 内容服务器还确保了销售过程的安全。

7.2.4 方正 Apabi 电子书系统

1. 背景

2001 年年初，电子工业出版社率先与北大方正结成战略联盟，将出版社的知识资源体系与北大方正开发的 Apabi 电子书整体解决方案有机整合，正式进军网络出版市场。2002 年适逢电子工业出版社成立 20 周年，在 20 周年社庆到来之际，电子工业出版社与北大方正加强了合作深度，联合发起了“千本电子图书迎接 20 周年社庆”活动。

虽然电子书的制作周期相对比较快，但是短期内迅速完成千本电子书，存在一定的难度。然而电子工业出版社的电子书制作人员胸怀满腔热情，他们尽可能排除各种困难，对电子书的制作倾注了全部心力，千本电子书制作完成了，为 20 周年的社庆献上了一份厚礼。同时电子工业出版社开通了电子图书的宣传、销售网站，这标志着电子工业出版社全面进军网络出版，由单一纸介质图书出版向以网络、光盘、多媒体教育软件等为载体的网络出版，由平面出版向多层次、立体出版的跨越。

北大方正推出了基于数字版权保护的中文网络出版整体解决方案——方正阿帕比 (Apabi)，Apabi 分别代表着 Author (作者)、Publisher (出版者)、Artery (流通渠道)、Buyer (读者，即购买者) 以及 Internet (网络)。作者、出版社、发行商和读者是传统出版产业链的有机组成部分，也就是说，Apabi 是以因特网为纽带，将传统出版的供应链有机地连接起来，实现完全数字化的出版。方正阿帕比数字版权保护系统 (Apabi DRM) 获得了 2003 年信息产业部重大技术发明奖，基于此技术的电子书解决方案称为 Apabi，最早的电子书解决方案是 Apabi 网络出版 (eBook) 系列解决方案，提供对个人读者的服务，读者通过在线支付后即可下载阅读电子图书。

电子工业出版社自从与北大方正合作以来，目前已经出版电子书一千余本，通过方正的数字图书馆系统、网站、电子书手持阅读器等多种渠道为读者所浏览，该社的电子书深受广大读者的喜爱，在网站上被读者多次下载，随着图书馆对电子书的需求越来越多，电子工业出版社的电子书也被众多图书馆所采购。

电子书的销售，不仅为电子工业出版社创造了一种收益来源，也为纸书的销售起到了宣传和推动作用。现在电子工业出版社里新出版的纸书一般都会比较快的制作出电子书。电子工业出版社在网络出版事业的发展中已经留下了坚实的足印。

2. 方正 Apabi 电子书系统结构

(1) 方正阿帕比制作出版软件

制作出版软件包括两个部分,即方正阿帕比转换软件(Apabi Maker)和方正阿帕比编辑软件(Apabi Writer)。

Apabi Maker 是一个数据转换工具,可以把用于印制的电子文档,包括 S2、S72、PS2、PS、EPS、TIFF、DOC 等文件转换为 CEB (Chinese eBook, Apabi 文件格式)格式。转换好的 CEB 文件完全保持原书的版式,包括原书的字体、图片、表格、公式、色彩等复杂的版面内容。

Apabi Writer 主要针对出版社用来填充书目信息、制作图书目录链接,还可以制作电子书的源数据信息,如作者、书名、书号、定价、出版社和摘要等。同时,Apabi Writer 还承担了书库和图书馆的作用,在大量的图书当中,出版社可以快速地从中选取读者需要的书籍。

(2) 方正 Apabi 的安全和交易软件

安全和交易软件由 Apabi Rights Server 和 Apabi Retail Server 组成,Apabi Rights Server 提供给出版者,Apabi Retail Server 则为网上书店提供电子书的交易平台。经过 Apabi Maker 生成的 CEB 文件,可以提交到方正安全发行软件 Apabi Rights Server,它会自动对提交的 CEB 文件进行加密,保护电子书的数字版权。并管理相应的数据(即作者、书名、书号、定价、出版社名和摘要等信息),分配给不同角色以相应的权限,进行电子书的上载、发布和销售等。

实现电子书信息平台的管理。Apabi Rights Server 还提供了数据查询和数据管理的功能,可以管理用户、下载站点、订单,实现电子书多级分发,并提供精确的销售数量统计功能,为相关环节之间的商务结算提供基础,像印刷书一样把握电子书的销售。Apabi Retail Server 用在书店端服务器,用于电子书的销售统计,网上书店可以通过方正 Apabi Retail Server 得到电子书下载许可证,并发给读者。方正 Apabi Retail Server 会自动与相应的方正 Apabi Rights Server 交换信息,以确保版权的保护以及信息的安全传递等。

(3) Apabi Reader

Apabi Reader 包括阅读器、藏书阁和书店。它可以阅读 CEB、PDF、HTML、TXT 等文件。其主要功能有翻页、加批注、画线、加书签、查找等。通过 Apabi Reader 读者可以在网上购买、阅读和下载图书,建立自己的电子图书馆,实现图书的分类管理。

方正 Apabi 电子书系统的特点如下。

① 方正的电子书出版系统对于电子书产业链上的各个环节都有完整的支持,包括出版社制作图书的软件(Apabi Maker、Apabi Writer)、发布图书的系统(Apabi Rights Server)、电子书店销售图书的系统(Apabi Retail Server)、图书馆提供电子图书借阅服务的系统(Apabi Library Server)、读者阅读图书以及在手机上和在专用阅读器上阅读的软件(Apabi Reader)、出版社图书销售在线查询系统(Apabi eStat 交易统计中心)等电子书产业链上各种角色所需要的软件和系统。

② 在系统完整性方面领先于国外同行。

③ 方正的电子书 DRM 系统支持对个人和对图书馆都按“本”进行销售。

④ Apabi 针对中国国情,在国际上首次实现了出版社和电子书店的多对多模式,即多家

出版社的电子图书可以在同一家书店销售，而同一家出版社的图书可以在多家书店同时销售，并且出版社的图书不仅可以在电子书店销售，也可以在出版社自己的网站上开一个电子书专卖店进行销售。

3. 方正 Apabi 电子书解决方案

方正 Apabi 解决方案把电子书的流程还原为纸书的流程。通过方正 Apabi 产品，出版社可以很方便地制作和出版电子书，并且保护作者及出版社的版权。网上书店很容易建立网上电子书的销售系统。读者可以通过网上电子书店买书、在本地设备上阅读电子书，但购买的电子书只能通过授权传播。读者也可以在数字图书馆借书、还书。图 7-5 为整个流程的示意图^[1]。

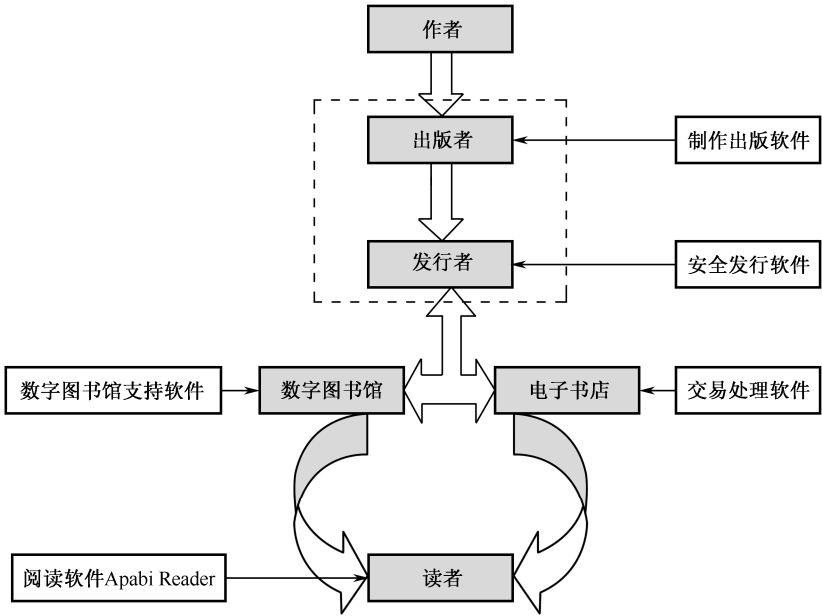


图 7-5 Apabi 电子书流程图

方正 Apabi 电子书解决方案，基本上还原了纸质书的商业模式。在此模式基础上，解决了电子书数字版权管理的一些基本问题，如：采用硬件信息相关的加密方法，限制电子书的二次传播，即防止对电子书的盗版；不同服务器之间的认证方法，确保电子书销售数据的准确性以及可审计性；在阅读软件中，限制文字的复制次数，避免对电子书全书的文字拷贝，又使读者能作适当的文字引用；读者可以借、还电子书，也可以把电子书送给其他读者等。

7.2.5 电子书 DRM 应用方案的比较分析

以上各电子书 DRM 方案可以保证交易和统计按照“本”来进行，按“本”销售是电子书产业界的一个趋势。通过加密、数字签名等技术手段，可以使得每销售一本电子图书都在中立的统计中心进行统计，并且每次的交易都有销售书店的数字签名，保证了每次交易都是不可抵赖的，从而使得整个数据统计过程都是可信的。DRM 使得电子书更像我们所熟悉的纸书了，从而能够保证电子书行业的可持续发展。

在系统的完整性和适用性方面，方正 Apabi 要领先于 Adobe 和微软，充分发挥其后发优势，以及在传统图书出版领域的背景优势，使得方正 Apabi 在电子书领域与国外同行发展同步。其产品也更适合中国国情。Apabi 的电子书系统通过数字版权保护技术对电子书进行保护，解决了出版社关心的电子书的安全问题。

表 7-11 对方正 Apabi、Microsoft 和 Adobe 的电子书 DRM 应用方案进行了比较^[1]。

表 7-11 电子书的 DRM 应用方案对比

品 牌	方正 Apabi	Microsoft	Adobe
密钥	168 位 DES 随机对称加密	1024 位 RSA 公钥加密	64 位和 128 位 RSA 双重密钥
个人用户	支持	支持	支持
数字图书馆	支持	不支持	支持
按“本”销售	支持	支持	支持
销售统计功能	支持	支持	支持
阅读器格式	CEB、PDF、HTML、TXT	Word、HTML	PDF
电子商务功能	支持	不支持	支持

7.3 电子文档的 DRM

电子文档作为传统纸质文档的信息化载体，具有成本低、使用易、通信快、发布广的优点，在企业信息化建设和政务信息化建设过程中越来越受到推崇。但是，在上述优点给企事业单位带来利益和效率的同时，电子文档信息容易获取的特性，也使很多企事业单位忧心电子文档的信息泄密风险。通过二次传播（E-mail、U 盘等）造成的信息扩散是电子文档泄密的主要途径。于是，越来越多的企事业单位希望利用 DRM 技术来保证电子文档信息的发布、流通和合理使用。

7.3.1 电子文档的格式

常见的基于 DRM 技术的电子文档格式主要有两大类型。

1. 非固定版式的文件格式

这方面的 DRM 产品，常见的有对 Office 文档的保护和对 HTML 格式的保护。例如微软的 Office 2003 就带有含 DRM 技术的 IRM 服务，保护对象是 Word、Excel、PowerPoint 文档。然而，由于非固定版式文件格式的可编辑特性在安全性上往往带来更多的缺陷和风险。同时，常用的非固定版式文件的浏览器都是相当流行的桌面软件，例如微软的 Office 系列软件和 IE 浏览器，这些软件为了支持其他各种各样的应用要求，公布的开放接口非常多，也带来了很多的安全漏洞和安全隐患。因此，在安全电子文档产品中，基于固定版式的产品往往占有较大的优势。

2. 固定版式的文件格式

版式文件是指版面排版比较规范的文件。其特点是在任何环境下阅读，其版式都不会变

化,而且通常版式文件的内容都是成型的内容,不再允许更改。常见的支持 DRM 应用的版式文件格式有 PDF、CEB、WDL 等。PDF 是目前市场上使用较多的版式文件。以 PDF 为格式的电子文档 DRM 产品中,最为著名的当属美国的 Authentica 公司的 Secure Documents for PDF 系统。此系统的核心采用 RC4 算法进行内容加密,使用 PDF 公开的 Plug-in 技术进行 PDF 文件控制,由 Policy Server 进行授权分配和管理,在英文版式市场上拥有较高的地位。

在中文版式方面,北大方正的 CEB 格式是一个比较突出的版式。CEB 格式在版式描述方面与 PDF 的能力比较一致,而且在 DRM 技术的基础上进行设计,在 DRM 体系方面有较好的支持,方正的 Aapbi 系列产品在电子图书、电子文档、电子政务等领域都支持 DRM 技术,目前在国内有较大范围的应用。例如 Apabi 重要文档防扩散系统(Apabi CEB DRM)产品就是针对电子文档的 DRM 产品。

7.3.2 基于 RMS 的 Microsoft Office 2003

■■■■

1. RMS 概述

Microsoft Office 2003 提供了信息版权管理(IRM),这种新功能有助于避免敏感信息落入没有权限的用户手中,无论发生意外或不慎造成的情况。可以授予用户阅读和更改的访问权限,并设定内容的有效期,也可以删除文档、工作簿或演示文稿的受限许可。此外,企业管理员可以创建可应用于 Microsoft Office Word 2003、Microsoft Office Excel 2003 和 Microsoft Office PowerPoint 2003 的许可策略。这些策略指定谁可以访问信息,以及对一个文档、工作簿或演示文稿,用户具有什么样的编辑级别或能力。

一般情况下,客户端计算机必须连接到公司的网络中才可以获得受 RMS 保护内容的发布许可。如果在客户端计算机未连接到公司网络的情况下使用这些计算机发布受 RM 保护的内容,则需要进行客户端注册,使用许可证书在未连接到公司网络的情况下发布受 RM 保护的内容。文档的作者可以使用 Microsoft Office 2003 来设置与企业的业务策略相一致的内容使用权限和条件。接收受 RM 保护内容的每个用户均可以通过 Windows RMS 请求和接收用户许可证,其中列出了该用户使用该内容时的使用权限和条件。Microsoft Office 2003 可以使用 Windows RM 技术来读取、解释和实施使用权限和条件。支持 RM 的应用程序使用对称密钥加密内容,所有 Windows RMS 服务器、客户端计算机和用户账户都具有相关联的 1024 位的 RSA 密钥。

2. 系统功能与特点

基于 RMS 的 Microsoft Office 2003 提供了以下功能。

① 企业中的 IT 管理人员可以为用户创建和分发文档,定义使用权限和条件的权限策略模板。例如,可以为员工创建权限策略模板,以便对公司的机密信息按照不同部门的访问能力单独分配使用权限和条件。对于那些要为其内容建立文档分类层次结构的组织而言,这些模板提供了一种便于管理的方法。

② Windows RMS 支持日志记录,管理员可以跟踪和审计组织内受 RM 保护内容的使用情况,以便记录 RM 的活动情况。

③ Microsoft Office 为商业环境中的信息工作者提供了客户端软件、服务器平台和相关服务，帮助企业成功地将信息转化为商业竞争力。

该系统具有以下特点。

① 持久的、跨网络、跨地域的文件级保护机制，保证文件的使用权限控制将一直紧紧跟随。

② 适用性强，且具有很好的兼容性。由于微软强大的产品市场实力以及广大的用户基础，所以其产品很容易得到社会的认可。

③ 应用范围广。由于 Word、Excel 等 Office 软件是各大公司必备的办公产品之一，可以说大家对它再熟悉不过了，在此基础上适时推出 DRM 功能，必将得到广泛关注。

④ 易用性较差。由于微软产品一贯的风格，注册和许可证机制过于烦琐，往往使用户失去耐心。

7.3.3 Adobe 公司的 Adobe Acrobat

在电子文档领域，Adobe 公司同样也推出了其 DRM 产品——Adobe Acrobat。

1. Adobe Acrobat 系统的功能与特点

Adobe Acrobat 6.0 支持作者和接收者签名，为电子文档的创建和其完整性提供额外的安全保证，通过使用新增的“Save as Certified Document”选项，作者可以在电子签名后，将文件，例如新闻稿件或分析报告，安全地传递给其他人。而接收者通过 Acrobat 或者免费的 Acrobat Reader 软件，也可以非常容易地验证那些被作者认证的文档签名，无须通过其他任何安全软件，就能够保证文档来源的可靠性，以及经创建以来的无改动性。

通过电子邮件或 Web 交换电子文档可以简化并加速信息交换。但是，若该信息为敏感或机密信息，必须锁定电子访问以确保隐私和安全。通过 Acrobat 和免费的 Acrobat Reader 软件，接收者也可以申请电子签名来对他们收到的信息予以答复，批准并回复给作者，此项功能特别适合于电子表格。通过使用免费的 Acrobat Reader，接收者也可以在线或脱机填写表格，并加上电子签名，回复给作者。举例而言，认证的文档和接收者的签名可以在一个完整循环的工作流中一起使用，从而建立一套完整可审计的文档跟踪流程。表格的最后接收者能够证实原始表格的真实性与完整性，而全部信息均可在工作流程内提供。

该系统具有以下特点。

① 提供灵活的权限管理功能，以控制对受保护的 Adobe PDF 文档的访问。为了更好地控制对文档的访问、修改和打印，作者也可以在一个单一的文档中对每个不同的申请用户授予不同的使用权限。

② PDF 文档具有的持久化安全特性，将针对不同使用权限的人员，保持文档访问和使用权限的控制，一个受保护的文档可以传递到任何地方。

③ 在网络内外、在线和脱机的情况下均能维护对文档的控制。

④ 高效地保护和分发机密信息，为共享密码指定嵌入式访问权限，该权限准确地定义了特定个人有权和无权对文档进行的操作。

2. Adobe Acrobat 的安全性能

由 PDF 格式标准提供的内置加密方法有下述功能：一个文档有两个密码，一个是文档所

有者（指 PDF 文档的创建者）密码，另一个是用户（PDF 文档的使用者）密码。一个文档也定义了即使该文档已被解密时应该被限制的操作，包括打印、将文本和图形复制出文档、修改文档以及增加或修改文本注解。在输入了正确的用户密码后，该 PDF 文档被打开并被解密，但上述操作仍然受限制：只有当输入 PDF 文档的所有者密码后才允许所有的操作。要改变所有者密码、用户密码以及取消操作限制，需要所有者密码。无论是对文档加用户密码、所有者密码或操作限制，由于加密位多达 40bit，所以其安全性很高，使用者不能轻易下载或复制，有力地保护了知识产权。图 7-6 为 PDF 文档安全设置。

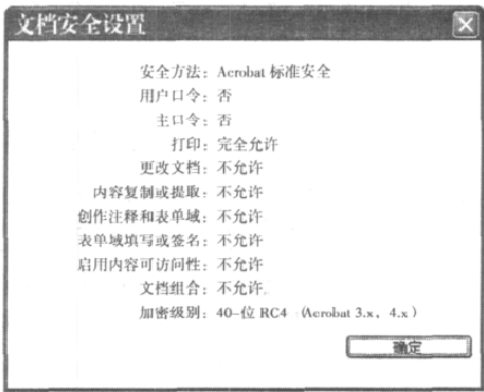


图 7-6 PDF 文档安全设置

7.3.4 北大方正 Apabi 文档保护系统

方正 Apabi 文档保护系统^[7]是一个可控授权下的文档安全共享系统，采用 DRM 技术，结合特有的安全文件格式，为组织的电子文档提供内容级的安全保护，防止因人员离职、恶意复制或病毒影响等因素造成组织的重要数据文件扩散或者泄露。

1. Apabi 文档保护系统构成

方正 Apabi 文档保护系统由以下几部分构成。

(1) Apabi 转换软件（Apabi Maker）

这是数据转换工具，可把 DOC、WPS、PDF 等各种格式的电子文件，转换为用于发布的 CEB 文件，并完全保持原文的版式，CEB 文件具有防篡改的功能，能在后台转换 Office 文档。

(2) Apabi DRM 服务器

采用 DRM 技术，通过文档加密、密钥管理、文档使用权限控制、硬件绑定管理等功能，对 CEB 文档进行安全保护。

(3) Apabi 阅读软件（Apabi Reader）

该软件用于 CEB 文档的阅读，以及与 Apabi DRM 服务器通信以获得对应的文档权限，控制文档的阅读、打印、打印份数等信息。其可嵌入 IE 中使用。

图 7-7 为 Apabi 文档保护系统的结构图^[4,7]。

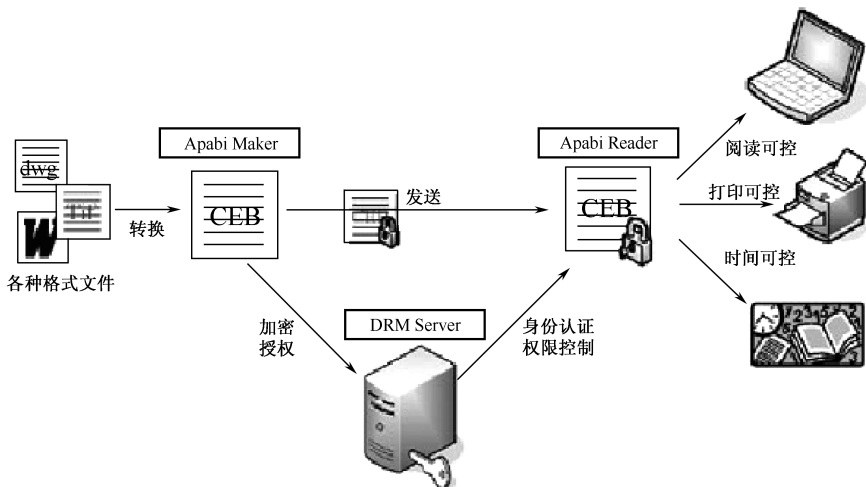


图 7-7 Apabi文档保护系统结构图

Apabi 文档保护系统的特点如下。

（1）信息安全保障方面

① 文档内容本身的加密，用户与 DRM 服务器通信加密，数据库关键字段的加密，这三重加密保证重要文档的内容安全、传输安全、数据库的安全。

② 文档加密采用 192 位的对称加密算法，即使文档被泄露出去，也不能被打开。密钥完全随机生成，每个文档的密钥不同，没有任何简单的破解办法。

③ 权限可控。不同的人对同一文档有不同的使用权限，能控制文档的阅读、打印、摘录、打印份数等权限；设定文档的有效访问时间，可以“召回”某些人的权限。

④ 控制使用者的截屏操作，防止截屏软件拷贝屏幕造成信息泄密。

⑤ 身份认证。支持 PKI 体系，采用用户名、密码、机器绑定等综合信息进行认证，文档只能被合法的用户打开，防止二次传播。

⑥ 采用信息摘要，防止对电子文档内容和权限的篡改。

⑦ 追踪文档的使用过程。记录受保护文档的使用情况（如文档在何时、被何人、在哪台计算机上、做了何种操作），追踪非法访问记录，为泄密事件提供追查依据。

（2）使用方便

① 模板功能：模板是一组权限的集合，将文档的不同访问权限赋予不同的人员，这些权限控制的集合保存成一个模板，这样对文档授权时，选择模板，系统自动将一组权限的集合授予文档，方便授权过程中的操作。

② 目录功能：同一目录下的文档享有相同的权限，由权限管理员设定好目录权限后，通过本系统上传到该目录下的文件自动授权。目录可以是虚拟目录（网页上的栏目）和实际的目录（文件服务器上的共享目录）。

③ 批处理工具：将一批文档自动加密、授权、上载到授权目录下，这批文档自动具有授权目录的权限设置。

④ 在用户管理方面，支持用户、组、域用户，方便了用户管理。

⑤ 在二次开发支持方面，提供了完善的开发接口，以及完整的开发实例（上层子系统），使整个系统可以作为一个功能模块集成到任何应用系统中，支持 IE 直接浏览。

(3) 适应性强

在信息系统中，系统管理员往往拥有最大的权限，而该系统中，系统管理员与业务人员的权限是分离的，与现有的管理方式一致，通过对重要文档的不同使用权限可以与目前实际的管理方法进行有效结合。本系统既可以单独使用，也可以作为一个开发平台、应用系统的一个功能模块。

(4) 有效降低企业运营风险

该系统在信息化系统中起到了核心信息的保护作用，能够防止员工或合作伙伴通过二次传播手段将企业的商业秘密透露给竞争对手，从而降低企业因为泄密造成的商业风险，增加企业和员工、企业和合作伙伴之间的信任。

2. 使用流程

Apabi 文档保护系统的使用流程从逻辑上可分为：加密、授权、发送、使用这几个环节，加密是系统后台自动完成的，在授权环节，也可以使用目录功能在后台自动完成。图 7-8^[7]描述了 Apabi 文档保护系统的一个使用流程。在使用流程中，把使用者分为两种：文档作者和文档使用者。

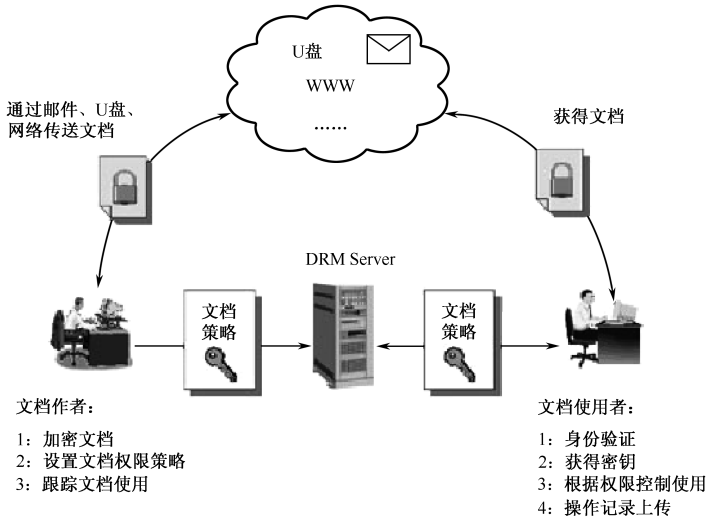


图 7-8 Apabi 文档保护系统使用流程图

用户在访问受保护文件前必须首先得到服务器的权限和身份验证，验证通过才能获得密钥和访问权限，Apabi Reader 根据获得的密钥和权限对文档进行控制。如果文件不慎被泄露出去，由于文件被加密和得不到验证，文件将不能被打开。

7.5 开放源代码 OpenIPMP

OpenIPMP^[8]是一个开源的 DRM 框架，是基于 MPEG、ISMA 的开放标准和其他国际标准的。OpenIPMP 主要由四部分组成。

① 打包器

主要负责将原始的文件使用加密算法进行加密处理后得到受保护文件。同时将加密后的

媒体内容及其元数据注册到 DRM 管理平台中。

② DRM 管理平台

用户注册，主要负责系统用户的添加工作，此处的用户没有区分是管理员还是终端用户。用户输入注册基本信息后，管理平台将用户添加到本地数据库中，同时将用户注册到 CA 中心数据库中，然后使用 768 位非对称加密算法 RSA 创建公钥和私钥，接着到 EJBCA 中心申请 X.509 证书，使用创建的 RSA 公钥对证书进行签名，最后将用户信息、RSA 私钥以及签名证书保存到 PKCSI2 结构中。证书下载，主要负责将此注册用户所申请的证书以 PKCSI2 结构的方式存储到本地。内容注册，由打包器发出注册内容的请求，管理平台将内容以及元数据信息保存到数据库中，同时使用 DOI 的简单方式来为此注册内容分配一个主 DOI，同时也分配一个实例 DOI。内容授权，为注册的内容进行用户授权，在此只支持单用户、播放许可、时间约束的版权授权。许可发放，用户通过播放器播放的时候需要先到管理平台申请许可，管理平台通过用户名及内容到数据库中搜索用户是否有此内容的授权协议，如果有并且协议有效则发放许可。

③ 播放器

主要负责播放受保护的数字内容。用户首先要确认自己的身份，所以播放之前首先会通过本地的 PKCSI2 证书结构到 CA 中心去做用户认证，认证通过后会去管理平台申请播放许可证，拿到许可证后对数字内容进行解密播放。

④ CA 中心

主要负责用户证书的发放以及用户的认证授权工作。客户端组件和认证书管理服务之间的所有通信都采用安全通信方式。

OpenIPMP 的系统框架如图 7-9 所示^[9]。

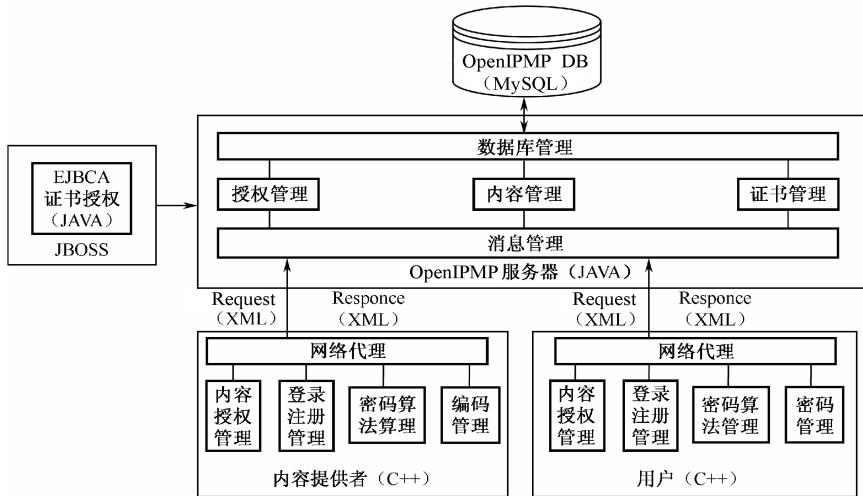


图 7-9 OpenIPMP 系统框架

OpenIPMP 的工作流程如图 7-10 所示^[10]。

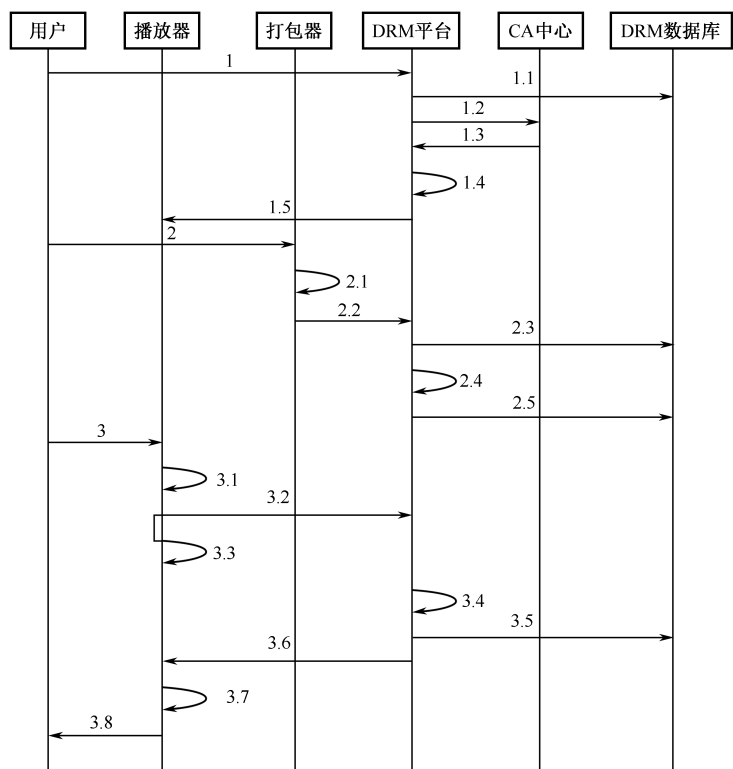


图 7-10 OpenIPMP工作流程

- 1. 用户通过管理平台注册，输入用户信息，包括用户名、密码、E-mail 地址等。
 - 1.1 DRM 管理平台将用户注册信息添加到平台数据库中。
 - 1.2 DRM 管理平台将用户信息注册到 CA 中心数据库中，同时通过 CA 中心得到一个 X.509 签名证书。
 - 1.3 CA 中心将用户名、密码及 RSA 公钥生成的 X.509 签名证书返回给 DRM 管理平台。
 - 1.4 DRM 管理平台将 RSA 密钥对、签名证书以及由用户名、RSA 私钥和签名证书构造的 PKCSI2 存储结构返回用户。
 - 1.5 用户将 PKCSI2 证书存储结构存放到播放器能够找到的地方。
- 2 用户通过打包器，输入打包者的用户证书以及元数据和媒体文件等信息。
 - 2.1 打包器利用加密算法对内容帧进行加密处理，得到受保护文件。
 - 2.2 打包器发送内容注册请求将受保护文件机器加密的密钥注册到 DRM 管理平台。
 - 2.3 DRM 管理平台将内容信息及密钥信息添加到 DRM 数据库中。
 - 2.4 用户通过 DRM 管理平台对注册的内容进行用户授权。
 - 2.5 DRM 管理平台将内容授权协议保存到 DRM 数据库中，以便以后发放许可。
- 3 用户利用播放器选择受保护的内容文件进行播放。
 - 3.1 播放器弹出用户认证页面，等用户输入了用户名及密码后，PKCSI2 会将证书里面的加密的用户信息利用 PKCSI2 中的私钥进行解密，然后与用户输入的用户名、密码进行对比认证。
 - 3.2 如果认证通过，即用户名和密码一致则向 DRM 管理平台发送获取许可的请求。

3.3 如果认证没有通过,则继续提示用户名和密码。

3.4 DRM 管理平台接收到用户获取许可请求,利用用户信息和要播放的内容信息在数据库中搜索此内容对此用户的授权协议。如果有授权协议,并且授权合法,则验证用户的证书是否合法,如果合法则用 ODRL 版权语言组织许可信息。

3.5 发放许可之前,DRM 管理平台将发放的许可信息保存到数据库中。

3.6 将许可信息发放给播放器。

3.7 播放器利用发放的许可,检查是否有当前动作的许可,如果有则用许可中提供的内容解密的密钥对内容进行解密播放。

3.8 用户观看内容。

参考文献

- [1] 王法涛. 数字权限管理技术 (DRM) 应用研究. 吉林大学硕士学位论文, 2006.
- [2] 焦亨. 流媒体 DRM 的研究与应用. 北京邮电大学硕士学位论文, 2007.
- [3] Windows Media Rights Manager 7.1 SDK, <http://technet.microsoft.com/>
- [4] 张玮. 基于开放源码的 DRM 系统分析与再设计. 北京邮电大学硕士论文, 2008.
- [5] EBX Working Group. The electronic book exchange system (EBX) Version 0.8.July 2000, Draft.<http://www.ebxwg.org/pdfs/spec.pdf>
- [6] 赵继海. 基于 DRM 技术的电子书服务模式的构建. 情报学报, 2002, 21 (3): 323-327.
- [7] www.apabi.com
- [8] sourceforge.net/projects/openipmp
- [9] 张永乐. 开源 DRM 系统的逆向分析与再工程. 北京邮电大学硕士学位论文, 2008.
- [10] OpenIPMP 业务模式简介. www.csdn.net

印刷品防伪技术

随着信息技术和高质量图像输入输出设备的发展，特别是高精度彩色喷墨、激光打印机和高精度扫描仪、复印机的出现，各种印刷品的伪造及篡改变得更加容易，印刷品的盗版行为也日益严重。盗版者通过扫描已发行的正版产品，然后再进行打印或印刷、销售，来实现盗版。

常用的传统印刷品防伪技术有防伪油墨、防伪纸张、防伪不干胶微缩文字印刷、票据特种防伪印刷、安全线、加密技术、激光全息转移技术、定位烫印、电话电码防伪、原子核双卡防伪、防伪标识以及手工雕刻凹版印刷等。利用传统的印刷防伪技术虽然可以解决产品版权保护及防伪问题，但大部分在应用过程中存在着许多弊端，如成本高、工艺复杂、易伪造和泄露核心技术、应用范围窄、适应性不够强等。

基于数字水印的印刷品防伪技术与传统的印刷品防伪技术不同，数字水印防伪技术对印刷设备没有特殊要求，在制版或打印过程中将数字水印信息加入，不用改变印刷材料和设备。与传统防伪技术相比，其优越性在于安全可靠，易分辨、易识别、检测提取易操作，难以伪造，适应性强，防伪成本低，无须增加固定投资和特殊材料，无须增加印刷成本和改变生产工艺流程，即可将现有的包装、设计改良为高度保密设计，实现高技术防伪及版权保护。但目前的数字水印技术研究大多针对的是数字化信息，由于印刷图像与数字图像的特性以及传播途径的差异，使得传统的数字水印技术不能直接应用于印刷行业中。

如何将数字水印技术应用到印刷品中是一个重要的课题。随着证件、有价证券、艺术品等印刷品防伪的需求增加，数字水印在印刷防伪领域中的应用有很大的发展潜力，蕴含着巨大的市场潜力。如果能在数字作品打印前嵌入一定量的信息，从而产生能够抵抗打印扫描的数字作品，并且嵌入的识别标识信息在数字作品打印扫描后仍能准确地提取出，那么像护照、驾驶证、证书、身份证明文件这类的印刷证件将能得到极好的保护。

美国 Digimarc 公司率先推出了世界上第一个商用数字水印软件，而后又以插件形式将该软件集成到 Adobe Photoshop 和 Corel Draw 图像处理软件中。其嵌入的水印能在一定程度上抵抗打印扫描，但加入的水印内容非常有限，仅仅限于 Digimarc ID、Distributor ID 和 Copyright Year，且对于几何变换非常敏感。

8.1 抵抗硬复制输出的数字水印技术

抵抗硬复制输出的数字水印技术是数字水印算法的一个分支，它是从广泛的应用需求中产生的。数字水印作品经打印或复印等方式输出到纸上，以及通过扫描从纸上输入计算机，这些过程都称为硬复制。将数字水印技术用于硬拷贝图文的防伪，称为硬复制数字水印技术，它能够抵抗打印机、印刷机、扫描仪等输入输出设备的攻击。

由于特定的应用需求，抵抗硬复制输出的数字水印技术有着其自身的特点。从水印内容上讲，这种数字水印技术嵌入的水印内容应该是有意义的，可以提供出版物的版权信息等，以满足版权认证与保护的需求；从水印特性上讲，嵌入的水印应该是鲁棒性水印，以保证当载体图像受到各类攻击后仍然可以提取出有效水印信息；从水印的检测方法上讲，为了满足应用需求，水印应该是盲水印，检测水印时不需要原始信息的支持。

印刷行业出版物中水印嵌入和提取的整个过程如图 8-1 所示^[1]。

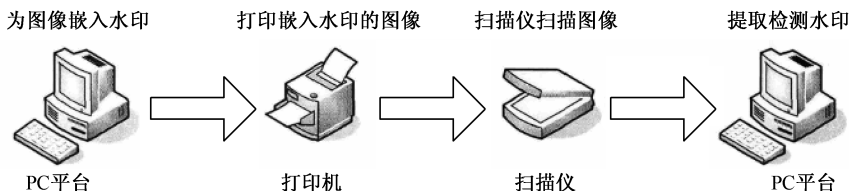


图 8-1 印刷行业中水印嵌入和提取的过程

数字图像在经过打印机打印时，打印机对图像进行了半色调处理，将图像由灰度图像转化为二值图像。数字水印的嵌入可以在半色调处理过程的不同阶段进行，即图 8-1 中的第一阶段可以分为三种情况，即半色调前嵌入水印、半色调过程中嵌入水印、半色调后嵌入水印。其处理过程如图 8-2 所示^[1]。

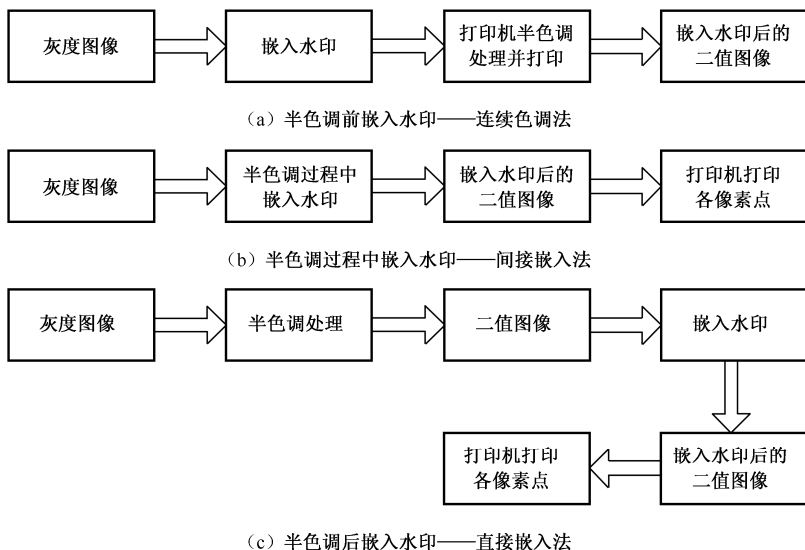


图 8-2 嵌入水印的三类方法

要利用水印对印刷品进行防伪认证,就必须对其经过打印扫描后的图像进行水印检测,打印扫描过程是一个复杂的 D/A、A/D 转换过程,在该过程中还会伴随着旋转、缩放、剪切等一系列的几何变换和像素失真,会对图像造成相当大的损伤。即经过打印扫描后,虽然得到一个视觉上和原图相似的复制品,但是,经过了一些扭曲失真,复制图像可能会丢失原数字图像的版权内容。

寻找合适的水印算法,使得数字图像在由印刷、打印输出后局部细节上失真,使用扫描仪进行扫描过程中造成图像畸变的情况下,仍然能够检测到所嵌入的水印,这是利用水印进行印刷防伪的关键,国内外的一些学者也开展了这方面的研究。

Ching-Yung Lin 等人^[2]较早开展了抵抗印刷扫描的水印算法的研究,对打印扫描过程进行了建模,分析了扫描后图像的特点,着重分析了图像 DFT 系数的变化,并提出了一种数学模型来描述图像经打印、扫描后的畸变。基于以上分析,找出了扫描后图像与原图之间的不变量,提出了一种基于 Fourier-Mellin 变换的能抗打印扫描过程的数字水印算法。

2000 年,Allebach 和 Kache 等人以扩频水印技术与直接二元搜索半色调技术为基础,提出了一种在二值图像中隐藏水印信息的方法,达到了图像加密目的。但该方法在透明性、稳健性两个方面均不理想^[3]。

2002 年,M. S. Fu 和 O. C. Au 在文献[4]中提出利用纠错码技术预处理水印信息,然后再以伪随机方式进行嵌入的 5 种半色调图像数字水印方案:DHST、DHPT、DHSPT、DHED 和 MDHED,在一定程度上提高了数字水印的透明性和稳健性。但纠错码的引入严重限制了可嵌入的水印信息容量,而且纠错码只能在一定范围内进行纠错,一旦错误码超出该范围,其纠错功能便会失效,也就是说该方法只能抵抗较弱的攻击。

2002 年,文献[5]提出了一种基于条件概率的半色调数字水印算法,该方法为半色调及数字水印技术提供了一条新思路,但其所产生的半色调图像质量较差(特别是纹理图像),且抵抗常见图像攻击的能力不强(特别是打印-扫描攻击)。

2004 年,文献[6]提出了一种基于随机误差分散的半色调数字水印方法,但该方法在提取水印信息时需要原始载体,而且其无法有效抵抗打印扫描等攻击。

2004 年,文献[7]结合 DCT 域中频系数比较,提出了一种半色调图像数据隐藏算法,并进行了抗打印扫描攻击实验。但该算法允许嵌入的数据量有限(56 比特)。

2005 年,Soo-Chang P.针对有序抖动图像提出了 PSMOD (Paired Sub-image Matching Ordered Dithering) 算法^[8],该算法用大小相间的有序抖动模板对半色调图像进行位交错预处理,得到明暗相间的子图,通过调整明暗子图对的顺序来嵌入二值水印。该算法利用了区域统计特性,对涂改、剪裁有很好的鲁棒性。

针对二值化图像的特点,研究人员提出了空域像素翻转的方法。其中,2004 年,M. Wu 等人给出了一个有效的水印算法^[9],根据二值图像像素的连通性和平滑性来对每个像素的可翻转性进行评分。2007 年,Yang 等人^[10]从二值化图像连通域的角度出发,详细分析了像素的翻转性,所提出的算法具有较好的性能。

8.2 打印扫描过程中图像的畸变分析

8.2.1 像素点的失真分析

像素失真主要包括亮度、对比度的变化, gamma 修正, 图像的半色调, 色彩的变化及相邻像素的模糊化。造成像素失真主要是由于打印(D/A 转换)过程中数字图像的半色调处理、扫描(A/D 转换)过程中扫描仪的工作原理以及同一图像在不同设备上具有不同色彩空间描述方式的色差失真造成的。

普通的激光打印机都属于点阵打印机, 对灰度图像而言, 它采用半色调技术, 用黑白点阵打印, 通常激光打印机常用的半色调技术是抖动法, 图像都具有一定的纹理, 在图像的任何一个小局部, 纹理往往是相似的, 像素灰度也是相近的, 而人眼视觉又具有空间低通特性, 抖动法正是利用这个特点, 用一组打印点代表相同个数的一组像素, 并通过这组打印点综合出这些像素的整体灰度效果。一般来说, 抖动法输出的图像大体上和原图一致, 局部细节上往往有失真, 常见的情况是在图像上出现高频抖动的纹理; 其次, 由半色调复合点的形状、激光束的扩散、纸张的吸水特性和光滑度等因素造成的半色调复合点变换(包括复合点增益和曝光), 也常会导致输出图像变得模糊不清。打印分辨率决定了图像每英寸上的打印点数, 通常分辨率越高, 输出图像质量越好。

扫描仪的质量取决于扫描仪中传感器的性能及 A/D 转换器对传感器输出电压的采样能力。由于人眼与电子元件对光线的反应关系不同, 对图像的感知也有所差别。比如说, 在图像的深色阴暗部分, 人眼能够清楚分辨的不同点, 从扫描仪输出的结果则可能是相同的。且传感器每个单元的光灵敏度不会完全相同, 它们之间的电隔离以及环境噪声也在一定程度上影响着传感器的工作性能, 还有电子电路本身的不稳定使扫描过程不可避免地引入噪声。因此, 几乎所有的扫描仪都会不同程度的给图像带来以下几个方面的问题。

- ① 整体反差失真。这是由于平板式扫描仪会混淆图像的动态范围, 压缩了影像明暗差距。
- ② 影像偏色。由于光学系统等方面的原因, 用扫描仪获取的图像极易发生偏色现象, 使得生成的图像色彩不正, 与原始图像有差别。
- ③ 网纹现象。扫描打印输出的图像时, 由于打印是以网点形式打印的, 因此在扫描时, 这些网点会原形毕露。
- ④ 聚焦不准。图像会看起来不够清晰。

下面通过实验分析打印扫描过程对图像像素失真的影响。直方图是像素按灰度变化的分布图(横坐标代表灰度, 灰度值由 0 至 255, 纵坐标代表像素个数), 它直观地反映了图像灰度分布情况。对于灰度图像, 其像素值在 0~255 之间变化, 因而可以将一幅 Lena 灰度图像[图 8-3(a)]的所有像素值分为 256 类, 通过统计 256×256 的 Lena 灰度图像的像素值中每类的个数, 得到其像素值分布的直方图, 如图 8-3(b)所示。为进行对比, 将 Lena 灰度图像打印扫描后重新得到了数字图像[图 8-3(c)], 其像素值分布的直方图如图 8-3(d)所示。

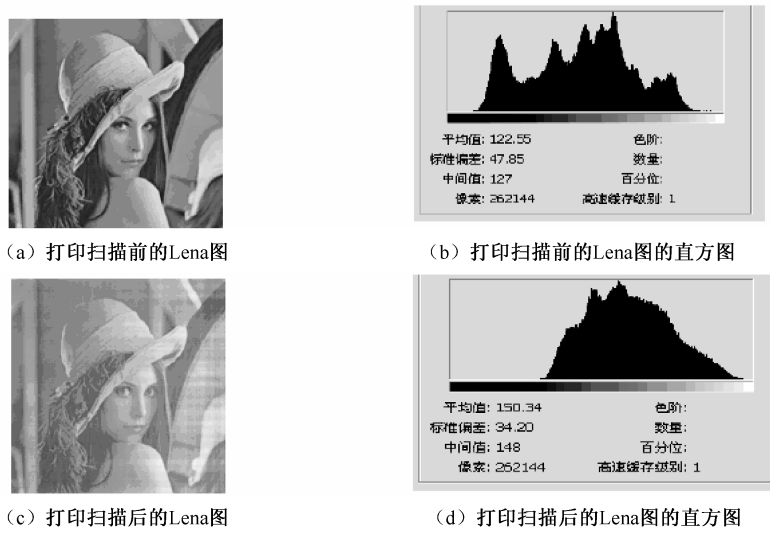


图 8-3 打印扫描前后Lena灰度图像的直方分布图比较

比较图 8-3 (a) 和图 8-3 (c) 可看出，打印扫描过程后的图像对比度和整体亮度降低，图片的细节部分已模糊不清。从图 8-3 (b) 和图 8-3 (d) 的 Lena 图像的像素值分布上可以看出，打印扫描后图像的直方图只是大体上与原始图像相似，但灰度分布发生了很大变化，主要是灰度级的范围缩小了，并且图像的像素值较多地分布在中等灰度区域。这与图 8-3 (a) 和图 8-3 (c) 所看到的图像整体较暗的结果一致。这种较大的差异将会导致水印检测的失败。

8.2.2 几何失真

从扫描的整个过程看，除了仪器的质量因素外，人为因素也很重要。若打印图片在扫描仪平板上放置时发生偏斜，则最后得到的图像可看成原始图像经过旋转、缩放和剪切这一系列攻击后的失真版本。另外，扫描过程的不均匀则会引起图像的非均匀几何变形。

相对于像素失真来说，几何失真更难处理，因为几何失真会使水印信息和载体失去同步，很多鲁棒性算法在几何攻击下都会失效，其表现为水印实际上还存在于图像中，但水印检测器已不能检测水印的存在。寻找抗几何失真的有效方法是当前水印研究的热点。对抗几何攻击的数字水印算法将在第 9 章进行详细讨论，在此不再赘述。

8.3 基于频域系数的抗打印扫描水印算法

基于频域系数的抗打印扫描算法是在半色调前嵌入水印。

Ching-Yung Lin 等人^[2]利用 DFT 变换和 Fourier-Mellin 变换，较早开展了基于频域系数的抗印刷扫描水印算法的研究。2004 年，牛少彰等人结合 DCT 域中频系数比较，提出了一种半色调图像数据隐藏算法^[7]。

本节介绍一种基于 DCT 域的自适应抗打印扫描算法^[11]。

8.3.1 打印扫描在 DCT 域上对图像的影响

为了分析打印扫描在 DCT 域上对图像的影响,将 $M \times M$ 的灰度图像按照 $N \times N$ 进行分块,可以得到 l ($l = M^2 / N^2$) 个块,对每个块进行 DCT 变换,得到 DCT 系数矩阵。将每一块的 DCT 系数按 ZigZag 顺序扫描为一维序列, ZigZag 扫描顺序如图 8-4 所示。将每一块中同一序号的 DCT 系数归为一类,这样 $M \times M$ 的灰度图像就分成了 N^2 个系数类,每一类可表示为 $C(u) = \{C_i(u), i = 1, 2, \dots, l\}$, $u = 1, 2, \dots, N^2$, 则 $C(1)$ 为 DC 系数类,其余均为 AC 系数类。

1	2	6	7	15	16	28	29	45	46	66	67	91	92	120	121	153	154
3	5	8	14	17	27	30	44	47	65	68	90	93	119	122	152	155	188
4	9	13	18	26	31	43	48	64	69	89	94	118	123	151	156	187	189
10	12	19	25	32	42	49	63	70	88	95	117	124	150	157	186	190	219
11	20	24	33	41	50	62	71	87	96	116	125	149	158	185	191	218	220
21	23	34	40	51	61	72	86	97	115	126	148	159	184	192	217	221	146
22	35	39	52	60	73	85	98	114	127	147	160	183	193	216	222	245	247
36	38	53	59	74	84	99	113	128	146	161	182	194	215	223	244	248	269
37	54	58	75	83	100	112	129	145	162	181	195	214	224	243	249	268	270
55	57	76	82	101	111	130	144	163	180	196	213	225	242	250	267	271	288
56	77	81	102	110	131	143	164	179	197	212	226	241	251	266	272	287	289
78	80	103	109	132	142	165	178	198	211	227	240	252	265	273	286	290	303
79	104	108	133	141	166	177	199	210	228	239	253	264	274	285	291	302	304
105	107	134	140	167	176	200	209	229	238	254	263	275	284	292	301	305	315
106	135	139	168	175	201	207	230	237	255	262	276	283	293	300	306	313	315
136	138	169	174	202	207	231	236	256	261	277	282	294	299	307	312	316	321
137	170	173	203	206	232	235	257	260	278	281	295	298	308	311	317	320	322
171	172	204	205	233	234	258	259	279	280	296	297	309	310	318	319	323	324

图 8-4 18×18 的 ZigZag 扫描顺序表

在引入上面的分类后,就可以比较每个系数类在打印扫描前后变化的情况。以 256×256 的 Lena 灰度图像为例分析。因为图像的边缘部分在扫描时可能和背景有一部分融合而造成像素偏差,为了更准确地分析,需要去除打印扫描前后图像的边缘部分,这里去除图像边缘的两个像素,也就是得到大小为 252×252 的图像。将其分成 18×18 的块,共有 196 块。分别在每一块的低频、中频和高频区域各取一个系数类来分析其打印扫描前后的变化。

在低频区域选取系数类 $C(13)$,对图像打印扫描前后 DCT 系数值对比分析,由图 8-5 可以看出这些系数只是幅值有所衰减,但系数的符号改变数量较少。

在中频区域选取系数类 $C(85)$,对图像打印扫描前后 DCT 系数值对比分析,由图 8-6 可以看出这些系数也是幅值有所衰减,形状比较相似,较大系数的符号改变数量也并不多。

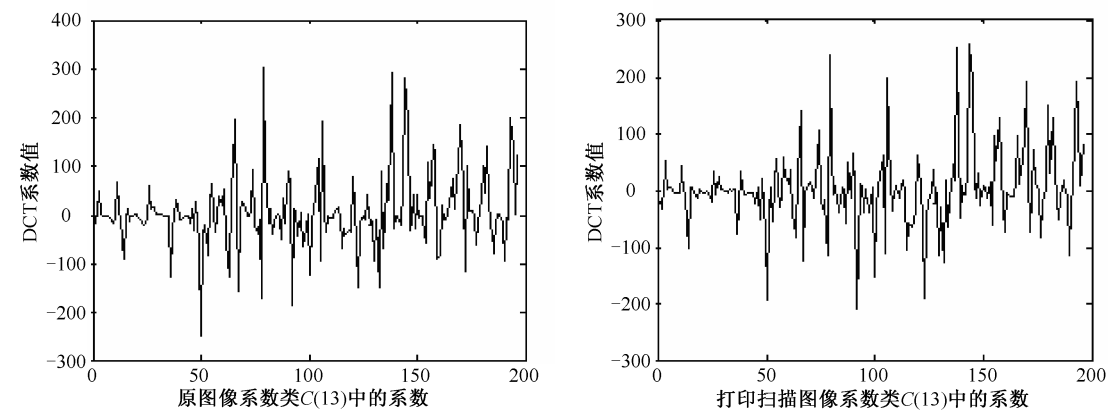


图 8-5 图像打印扫描前后低频区域系数类C(13)系数对比

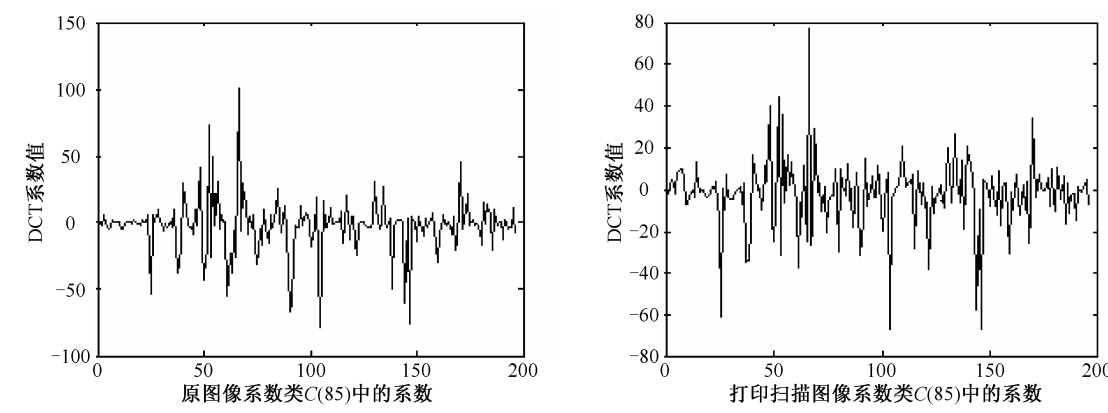


图 8-6 打印扫描前后图像中频系数类C(85)DCT系数对比

在高频区域选取系数类 $C(293)$ ，对图像打印扫描前后 DCT 系数值进行对比分析，由图 8-7 可以看出这些系数值很小，符号改变的系数数量也是很多的。

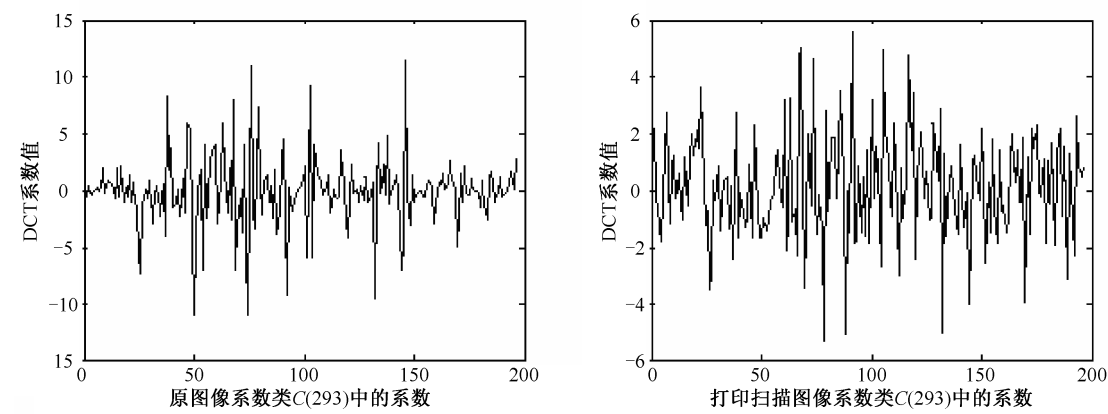


图 8-7 打印扫描前后图像高频区域系数类C(293)DCT系数对比

由于图像经打印扫描后像素值会产生失真，因此也就导致了由像素值计算出的 DCT 系数

值产生很大变化。但由以上的分析发现,中低频区域的系数类系数虽然幅值有一定程度的衰减,但很多值较大的系数的正负号几乎很少改变,因此可以利用这一特点,通过改变同一系数类中系数的正负号来表达水印信息。

下面具体分析一下每个系数类中的系数在图像打印扫描前后符号改变的情况。图 8-8 显示了每一系数类中在打印扫描前后正负号改变的 DCT 系数个数。可以看出从低频区域到高频区域,系数符号的改变数量逐渐增多,但不是稳定逐渐增多,而是出现许多小的峰值,也就是某些系数类的符号改变量比其相邻区域的要多。经仔细分析数据后发现,这些峰值出现的类恰好是每一分块的边缘区域,而靠中部的系数类符号改变的数量相对较小,因此水印嵌入算法的设计需要结合这一特点来提高抵抗打印扫描的鲁棒性。

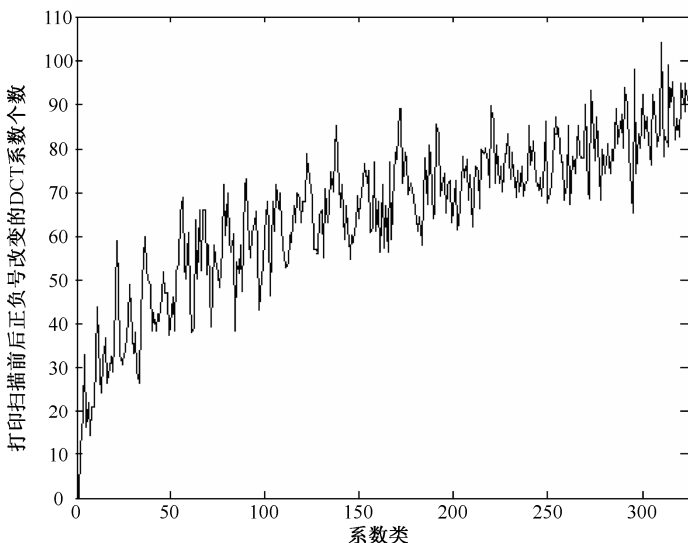


图 8-8 打印扫描前后图像 DCT 系数符号改变的情况

8.3.2 水印嵌入算法

根据上面的分析,利用图像 DCT 变换的中频系数符号的变化进行水印的嵌入。具体算法如下^[1]。

① 生成水印序列 $w = \{a_i \mid a_i \in (0,1), i=1,2,\dots, m\}$, 其中 m 为水印嵌入的比特数。水印可由密钥随机生成,也可由有意义数字转成二进制形式得到。

② 将 $M \times M$ 的灰度图像按照 $N \times N$ 进行分块,得到 $l(l=M^2/N^2)$ 个小块,对每个 $N \times N$ 的小块进行离散余弦变换,得到 DCT 系数矩阵。

③ 将每个块去掉边缘部分的两个系数,将剩余矩阵中的 DCT 系数按 ZigZag 顺序扫描为一维序列。将每块中同一序号的 DCT 系数组成一个系数类,每一类可表示为 $C(u) = \{C_i(u), i=1,2,\dots, l\}, u=1,2,\dots,(N-2)^2$ 。在其中选取中低频位置的系数类作为水印载体,对每一类系数的修改相当于嵌入 1bit 水印。

④ 将选中的系数类中的 DCT 系数按绝对值大小从小到大排序,记为 $P_s(u), s=1,2,\dots, l$, 设 j 为嵌入水印的起始位置, d 为水印嵌入的强度。

当要在 $C(u)$ 中嵌入 0, 即 $w(i)=0$, 修改系数:

$$P_s(u) = |P_s(u)|, s = j, j+1, \dots, j+d \quad (8-1)$$

当要在 $C(u)$ 中嵌入 1, 即 $w(i)=1$, 修改系数:

$$P_s(u) = -|P_s(u)|, s = j, j+1, \dots, j+d \quad (8-2)$$

注意到要在 $C(u)$ 中嵌入 0 时, 要把 $P_s(u)$, $s = j, j+1, \dots, j+d$ 中所有的负系数都改为正值。如果没有改动系数符号之前, 负系数的个数比正系数的负数多很多的话, 则需要改动很多负系数的符号才能满足嵌入水印的要求, 这样大的改动量将会对图像的质量产生影响。而且如果嵌入的水印容量比较大, 那么这种情况发生的几率也就会更多, 这样就会造成图像比较明显的失真。因此, 采用水印嵌入自适应的方法来解决这样的问题。

结合图像自身 DCT 域系数的特点来做到嵌入强度的自适应, 这样在增大水印嵌入容量的同时, 能更好地做到鲁棒性和不可见性的折中。与此同时, 通过结合图像自身特征来嵌入水印, 可以得到唯一的嵌入密钥, 大大增加了水印系统的安全性。因此, 在修改系数前增加以下步骤。

⑤ 根据水印嵌入系数类的特征生成密钥。令密钥序列为

$$S = \{x_i | x_i \in (0,1), i = 1,2,\dots, m\} \quad (8-3)$$

根据要嵌入的水印序列及序列对应的系数类 $P_i(u)$, $i = j, j+1, \dots, j+d$ 中正负系数的个数来生成 S 序列。令

$N_+(u) = P_i(u)$, $i = j, j+1, \dots, j+d$ 中正数的个数;

$N_-(u) = P_i(u)$, $i = j, j+1, \dots, j+d$ 中负数的个数。

当 $w(i)=0$ 时:

若 $N_+(u) > N_-(u)$, 则 $S(x_i) = 1, i = 1,2,\dots, m$;

若 $N_+(u) < N_-(u)$ 则 $S(x_i) = 0, i = 1,2,\dots, m$ 。

当 $w(i)=1$ 时:

若 $N_+(u) > N_-(u)$, 则 $S(x_i) = 0, i = 1,2,\dots, m$;

若 $N_+(u) < N_-(u)$ 则 $S(x_i) = 1, i = 1,2,\dots, m$ 。

⑥ 把水印序列结合密钥嵌入图像的适当位置。水印的嵌入强度设为 d , 当要在 $C(u)$ 中嵌入 1 时:

若 $S(x_i) = 1, i = 1,2,\dots, m$, 则修改 $P_s(u) = |P_s(u)|, s = j, j+1, \dots, j+d$ 。

若 $S(x_i) = 0, i = 1,2,\dots, m$, 则修改 $P_s(u) = -|P_s(u)|, s = j, j+1, \dots, j+d$ 。

同样的方法, 当要在 $C(u)$ 中嵌入 0 时:

若 $S(x_i) = 1, i = 1,2,\dots, m$, 则修改 $P_s(u) = -|P_s(u)|, s = j, j+1, \dots, j+d$ 。

若 $S(x_i) = 0, i = 1,2,\dots, m$, 则修改 $P_s(u) = |P_s(u)|, s = j, j+1, \dots, j+d$ 。

⑦ 将图像各块进行反离散余弦变换, 再加上水印嵌入前去除掉的边缘两像素图像, 得到含水印的图像。

8.3.3 水印提取算法

数字水印的提取是嵌入的逆过程。对已经打印的图像进行扫描, 重新得到含水印的数字

图像。将图像去除边缘两个像素后进行分块，每块做离散余弦变换，得到 DCT 系数，然后根据嵌入水印的系数类 $C(u)$ 来计算每类系数中 $P_i(u)$, $i = j, j+1, \dots, j+d$ 的正负号个数：

若 $N_+(u) > N_-(u)$ ，则 $w'(i) = 1, i = 1, 2, \dots, m$ ；

若 $N_+(u) < N_-(u)$ ，则 $w'(i) = 0, i = 1, 2, \dots, m$ 。

最后将得到的 $w'(i)$ 结合水印的嵌入密钥 S 计算得到最终的水印序列 \tilde{w} 。

$$\tilde{w}(i) = \begin{cases} 0, & w'(i) = 0 \& S(i) = 1 \text{ 或 } w'(i) = 1 \& S(i) = 0 \\ 1, & w'(i) = 0 \& S(i) = 0 \text{ 或 } w'(i) = 1 \& S(i) = 1 \end{cases}, \quad i = 1, 2, \dots, m \quad (8-4)$$

8.3.4 实验结果及分析

对像素值大小为 256×256 的 Lena 图像按照 16×16 分块大小进行实验^[11]。令 $i + j = t$ ，首先在 $t = 12 \sim 18$ 的位置上嵌入水印信息。按照水印嵌入算法，可以嵌入 72 比特，选取 $d = 150$ ， d 为修改系数的个数。在图 8-9 中，图 8-9 (a) 为原始图像，图 8-9 (b) 为嵌入水印后的图像，图 8-9 (c) 为打印扫描后的图像。



图 8-9 基于 DCT 系数的抗打印扫描算法实验

从图 8-9 可以看到在嵌入强度较大的情况下，对于某些中频系数和低频系数的改变，都不会引起可觉察的变化。其中嵌入强度由修改系数的个数 d 决定， d 越大，修改个数越多，表示嵌入强度越强，对图像质量造成的影响越大。

8.4 数字半色调技术

如图 8-2 (b) 和图 8-2 (c) 所示，也可在半色调过程中或半色调后进行水印的嵌入。分析半色调图像的特性，对数字半色调过程进行加工和改变，才能得到有效的半色调图像的水印算法。因此，研究半色调算法是嵌入水印的基础。本节首先对数字半色调技术进行介绍，在后续的内容中将介绍几个典型的半色调图像水印算法。

8.4.1 半色调技术概述

半色调技术是指用少量的色彩将一幅连续色调图像（如灰度图像和彩色图像）量化为一幅二值图像或只有少数几种色彩的彩色图像，且量化后图像的视觉效果和原始图像相似。量化得到的图像称为半色调图像。一般印刷机、油墨打印机、激光打印机只有两种色彩或非常

有限的几种色彩，它们不能完全显示出连续色调图像所包含的全部色彩信息。此时就需要半色调技术来表现连续色调的效果。

19 世纪 60 年代，半色调技术的出现改善了活版印刷方法生成图像效果十分呆板的问题。William Fox Talbot 在 1852 年提出了连续色调效果可以由栅格化图像来模拟的理论。最早的商用半色调栅格出现在 1860 年，它通过排列图像的像素点分布来打印连续色调图像。到了 19 世纪 90 年代，发明家们发明出一种装有玻璃网格的相机，它可将图像分割成不同大小的小点来得到半色调图像。由于人眼具有低通滤波特性，在一定的距离观看这些小点的时候，人眼会把这些点看成连续色调的图像。而今半色调图像已经可由计算机生成，并且被广泛应用于彩色图像的打印上。

如何合理地安排像素点，用有限的色域范围来表现图像效果，这是半色调技术的关键。半色调技术不仅简单截取颜色归一到低分辨率所含的颜色区间，而且需要将量化误差均匀地分散到整幅图像里。简单截取出来的图像视觉效果很差。

半色调方法按照半色调图像中点的大小或点之间的距离分为调幅（Amplitude Modulated Screen, AM）半色调和调频（Frequency Modulated Screen, FM）半色调。调频半色调方法是通过改变相邻点的距离来表示不同的灰度等级，而点的大小是不变的；而调幅半色调方法是利用点的大小表示灰度等级，点之间的距离是相等的。

调频半色调处理后的图像基本没有低频颗粒，没有明显的人工纹理，但打印稳定性上表现不足；调幅半色调处理后的图像有比较严重的低频颗粒，而且有周期性的人工纹理产生，在彩色打印时，不同色调相互叠加，会出现莫尔条纹现象，但在打印稳定性方面表现良好。针对调幅半色调和调频半色调各自存在的优势和不足，L. L. Daniel 提出了混合半色调（Hybrid Halftoning）的思想^[12]。混合半色调是指在同一加网图像中既包含调频半色调也包含调幅半色调的方式。混合半色调技术是借鉴调幅和调频两种半色调技术，既体现了调频半色调的优势，又具有调幅半色调的稳定性和可操作性。

调幅加网是点聚集态的加网方法，通常采用点聚集态有序抖动（Clustered Dot Ordered Dither）法对待加网图像二值化。这种有序抖动利用一个抖动矩阵将灰度图像转换成二值图像，其优点是实现过程简单，计算复杂性小，输出图像与输入图像的点数相同；其缺点是丢失了大部分细节信息，图像效果最差，不能很好地再现图像。

调频加网采用点离散态网点技术，以网点数量的多少来表现阶调，利用单位面积大小相同的点出现的频率来反映图像的层次变化。它们能保持网点尺寸的一致，但会改变网点之间的距离。许多半色调技术所产生的网点都是离散无规则的分布且通常比聚集态的调幅网点要小。

主要的半色调方法有阈值抖动法、误差分散法、点分散法、噪声半色调法等。

8.4.2 阈值抖动法

阈值抖动法大体可以分成随机抖动和有序抖动^[13]两类。这两种算法都需要一个模板，模板一般为方阵。如果我们定义的模板大小为 $N \times N$ 个像素，则可表示 $N \times N + 1$ 个灰度级。灰度可以用一定比例的黑白点组成的区域表示，从而达到整体图像的灰度感。例如，在一个 2×2

个像素的模板中，它有 5 个灰度级（图 8-10）。

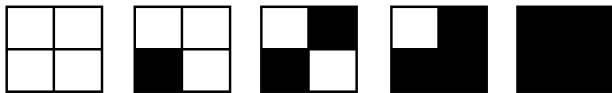


图 8-10 灰度级的表示

黑点分布可以是规则的，也可以是不规则的。一般情况下，有规则的模板比随机模板能够避免点的丛集，但有时会导致图像中有明显的线条。

模板的存储也是我们需要考虑的问题。例如，如果存储 256 级灰度的模板，就需要 $256 \times 16 \times 16$ 的二值点阵，占用的空间是相当可观的。更好的方法是：只存储一个整数矩阵，其中的每个值从 0 到 255。图像的实际灰度和矩阵中的每个值比较，当该值大于等于灰度时，对应点打一黑点。这个矩阵我们称为阈值矩阵。

我们用一个 25 级灰度的例子加以说明。图 8-11 左边为阈值矩阵，右边为灰度为 15 的图案，共有 10 个黑点，15 个白点。灰度为 0 时全是黑点，灰度每增加 1，减少一个黑点。

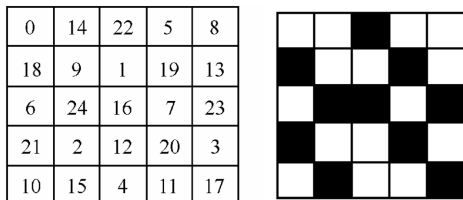


图 8-11 灰度级的表示

下面介绍一种设计阈值矩阵的算法，是 Limb 在 1969 年提出的。先从一个 2×2 的矩阵开始：

$$M_1 = \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix} \quad (8-5)$$

通过递归关系有

$$M_{n+1} = \begin{bmatrix} 4M_n & 4M_n + 2U_n \\ 4M_n + 3U_n & 4M_n + U_n \end{bmatrix} \quad (8-6)$$

M_n 和 U_n 均为 $2n \times 2n$ 的方阵， U_n 的所有元素都是 1。根据这个算法，可以得到 16 级灰度的阈值矩阵：

$$M_2 = \begin{bmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{bmatrix} \quad (8-7)$$

M_3 (8×8 阵) 比较特殊，称为 Bayer 抖动矩阵， M_4 是一个 16×16 的矩阵。抖动算法的效果主要取决于阈值矩阵的选取。各种抖动算法的本质不同就在于阈值矩阵的选择和构造方法不同，但是其算法的基本框架是大体相同的。

根据上面的算法，如果用 M_3 作为阈值矩阵，一个像素要用 8×8 的模板表示，则一幅 $N \times N$ 的图将变成 $8N \times 8N$ 大小。如果用 M_4 ，就变成 $16N \times 16N$ 了。能否在保持原图大小的情况下利

用模板技术呢？一种很自然的想法是：如果用 M_2 ，则将原图中每 8×8 个点中取一点，即重新采样，然后再应用模板技术，就能够保持原图大小。实际上，这种方法不可行。首先，不知这 8×8 个点中找哪一点比较合适，另外， 8×8 的间隔太大了，生成的图像和原图相差很大，就像图 8-12 中右边的那幅图一样。

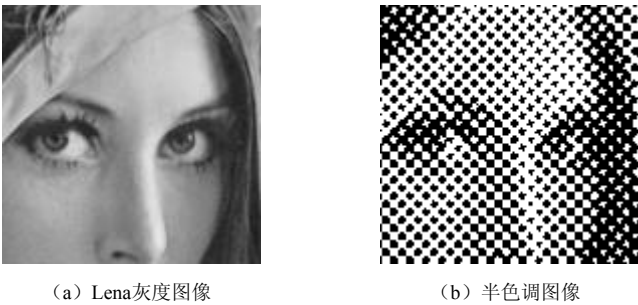


图 8-12 重采样后进行半色调处理

可以采用这样的做法：假设原图是 256 级灰度，利用 Bayer 抖动表，做如下处理：

if($g(x,y) \gg 2$) > bayer($y \& 7, x \& 7$) then 打一白点 else 打一黑点

其中， (x,y) 代表原图的像素坐标， $g(x,y)$ 代表该点灰度。首先将灰度右移两位，变成 64 级，然后将 x, y 做模 8 运算，找到 Bayer 矩阵中的对应点，两者做比较，根据上面给出的条件判断做处理。

可以看到，模 8 运算使得原图分成了一个 8×8 的小块，每个小块和 8×8 的 Bayer 矩阵相对应。小块中的每个点都参与了比较，这样就避免了上面提到的选点和采样间隔过大的问题。模 8 运算实质上是引入了随机成分，也就是抖动技术。

一般来说，设输入的灰度图像大小为 $M \times M$ ，输入图像各像素点的值为 $x(i,j)$ ，输出图像各像素点的值为 $y(i,j)$ ，阈值为 T 。阈值抖动的过程就是将阈值矩阵铺满整个输入图像，使得输入图像的每一个像素点都与阈值矩阵上的一个阈值相对应，把输入图像中每个像素点的值 $x(i,j)$ 和一个阈值 T 进行比较，按下式进行计算，得到输出的半色调图像。

$$\begin{cases} y(i,j) = 0, & y(i,j) < T \\ y(i,j) = 1, & y(i,j) \geq T \end{cases} \tag{8-8}$$

在随机抖动法中，阈值矩阵是一组随机数，随机数的取值范围在输入图像像素值的最小灰度到最大灰度值之间。有序抖动的阈值矩阵是有规律的，常用的阈值矩阵有两种——离散型和聚集型（图 8-13）。聚集型是指阈值大小从阈值矩阵某块区域向外逐渐增大，早期的打印设备或显示设备常使用聚集型，出来的效果往往是像素点之间没有明显的界限，造成图像的模糊，对图像细节信息的表现力不够，典型的如局部聚集整体分散。离散型指阈值离散分布到整个阈值矩阵。新式打印设备一般采用这种样式，能更精确地确定像素点位置和尺寸，图像模糊现象可以得到改善。较为常用的有 Bayer 有序抖动。

阈值抖动算法的优点是算法比较简单；但其缺点也很明显，使用阈值抖动算法得到的半色调图像图案化有时很明显。因为常用的阈值矩阵虽然引入了很多随机成分，但还是有规律的。其次，在阈值抖动算法的每个处理模块中，像素点的顺序是由阈值矩阵决定的，在算法执行过程中，这些像素点的顺序是固定的。因此即使阈值矩阵设计得非常好，输出的图像依

然会存在瑕疵。

0	32	8	40	2	34	10	42	42	43	44	45	46	47	48	49
18	16	56	24	50	18	58	26	41	20	21	22	23	24	25	50
12	44	4	36	14	46	6	38	40	19	6	7	8	9	26	51
60	28	52	20	60	30	54	22	39	18	5	0	1	10	27	52
3	35	11	43	1	33	9	44	30	17	4	3	2	11	20	52
51	19	59	27	49	17	57	25	37	16	15	14	13	12	29	54
15	47	7	39	13	45	5	37	36	35	34	33	32	31	30	55
63	31	55	23	61	29	53	21	63	63	61	60	59	58	57	56

(a) Bayer有序抖动阈值矩阵

(b) 聚集型抖动阈值矩阵

图 8-13 有序抖动阈值矩阵

例如，采用有序抖动的技术，在图像和阈值矩阵的点之间进行比较时，只要比阈值矩阵上点的值大就打白点，这种做法存在问题。如果阈值矩阵中某个阈值本身就很很小，而图像中点的灰度只比它大一点时则打白点，而图像中的点更接近黑色，而不是白色。一种更好的方法是将这个误差传播到邻近的像素。

阈值抖动算法存在着很多不足，下面介绍一种更好的半色调方法——误差分散法。

8.4.3 误差分散法

误差分散法^[14]的基本思想是将像素点半色调后产生的误差传播到邻近区域的像素点上，使得图像半色调所带来的误差在输出图像的整体效果中表现不明显。

误差分散法是一个自适应的算法，按照一定的扫描路径，对输入图像的像素逐个处理，计算其二值化后产生的误差，并将它分散到没有处理的相邻像素点上。分散过程中使用误差分散系数来控制各邻接像素点上分得的误差。当一个像素点被相邻点的误差修正后，它的像素值发生了变化，从而使得区域内的总体像素值的误差最小，输出的图像效果更优。典型的误差分散系统如图 8-14 所示。

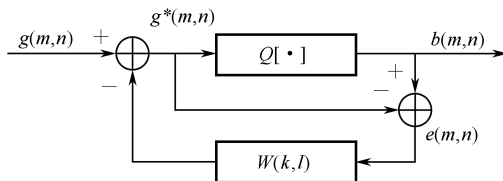


图 8-14 误差分散半色调系统

其中，输入 $g(m,n) \in [0,1]$ 表示原始的连续色调的图像，输出 $b(m,n) \in \{0,1\}$ 表示半色调图像，将图像看成一维信号。 $g^*(m,n)$ 为量化输入， $w(k,l)$ 为误差分散核的权重系数， $e(m,n)$ 为量化误差， $Q(\cdot)$ 为阈值量化函数，误差分散法的数学模型可用式 (8-8)、式 (8-9)、式 (8-10) 表示：

$$g^*(m,n) = g(m,n) - \sum_{k,l \in S} w(k,l)e(m-k,n-l) \quad (8-9)$$

$$b(m,n) = Q(g^*(m,n)) \quad (8-10)$$

$$e(m,n) = g^*(m,n) - b(m,n) \quad (8-11)$$

为了保证误差被全部分散，权系数的和为 1，即

$$\sum_{k,l \in S} w(k,l) = 1 \quad \text{且} \quad w(i,j) \geq 0 \tag{8-12}$$

阈值量化函数表示为

$$Q(x) = \begin{cases} 0, & x < 0.5 \\ 1, & x \geq 0.5 \end{cases} \tag{8-13}$$

误差分散半色调图像的视觉效果明显优于聚集型和分散型的阈值抖动图像，它没有周期性的人工纹理，而且将量化噪声分散到了人眼不太敏感的中频和高频段。如图 8-15 所示，图 8-15 (a) 是阈值抖动半色调法得到的二值图像，图 8-15 (b) 是 Floyd-Steinberg 误差分散半色调得到的二值图像，图 8-15 (a) 的效果明显劣于图 8-15 (b)，画面更加细腻。



(a) Lena 灰度图像 (b) 阈值抖动半色调图像 (c) Floyd-Steinberg 误差分散半色调图像

图 8-15 阈值抖动法与误差分散法的半色调图像

误差分散核的设计是实现高质量的误差分散半色调方法的关键。但是误差分散法容易在阴影区域、高亮度区域产生“虫状纹理”，这是因为误差分散核将量化误差向下、向右分散而造成的，采用增加误差的分散范围、调整误差分散的权重系数等方法可以有效减弱这种纹理现象。

Floyd-Steinberg 误差分散核是效果不错的四值误差分散核，Jarvis 和 Stucki 分别构造了有 12 个权重系数的误差分散核以减少误差分散图像中的“虫状纹理”，Hebery 也提出了有 25 个柯西系数的误差分散核，能完全去除“虫状纹理”。常用的误差分散核如图 8-16 所示，图 8-16 (a) ~图 8-16 (e) 五种误差分散核针对矩形栅格进行处理，误差分散核针对六边形的栅格进行处理，图 8-16 (f) 的误差分散核针对六边形的栅格进行处理。

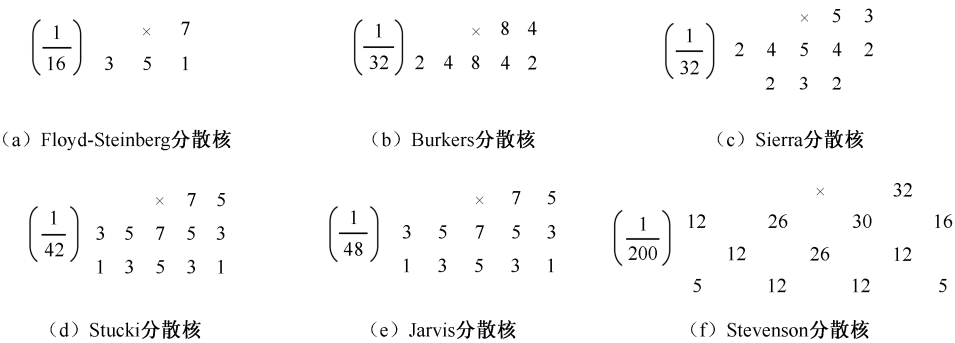


图 8-16 常用的误差分散核（其中×表示当前像素点）

8.4.4 点分散法

一种将有序抖动和误差分散的思想结合起来的算法被称为点分散法 (Dot-Diffusion)，由 D.E.Knuth 于 1987 年提出^[15]。它做到了大范围抖动，局部误差扩散。这样可使算法在提高图像质量的同时，做到并行处理。

点分散法涉及一个参数——等级矩阵 C (图 8-17)，该矩阵主要决定处理像素的顺序。设等级矩阵 C 的大小是 $N \times N$ ，元素的值从 1 到 $N \times N$ 。

35	49	41	33	30	16	24	32
43	59	57	54	22	6	8	11
51	63	62	46	14	2	8	19
39	47	55	38	26	18	10	27
29	15	23	31	36	50	42	34
21	5	7	12	44	60	58	53
13	1	4	20	52	64	61	45
25	17	9	28	40	48	56	37

图 8-17 Knuth点分散有序抖动模板

该算法的思想是从等级 $k=1$ 开始，然后增加等级，同时处理图像中等级 k 的所有像素，将它们与阈值进行比较，确定半色调图像相应像素的值。设输入的灰度图像大小为 $M \times M$ ，输入图像各像素点的灰度值为 $x(i, j)$ ，输出图像各像素点的值为 $y(i, j)$ ，阈值为 T ，按式 (8-8) 确定 $y(i, j)$ 的值。然后计算量化误差 $e(i, j) = x(i, j) - y(i, j)$ ，查看像素 (i, j) 的八邻域，调整其中等级高于 k 的像素 (未经处理的邻域)，将量化误差分散到等级高的 $x(i, j)$ 的相邻像素上。具体调整方法如图 8-18 所示。

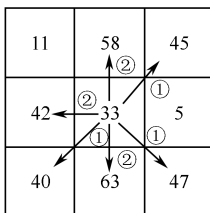


图 8-18 点分散原理

下一个等级 $k+1$ 对应的像素点按同样的方式处理。当所有等级处理完成后，算法结束。由于人的视觉系统对水平、垂直方向的误差较为敏感，所以在误差分配上水平、垂直比重大于对角元素，一般水平方向、垂直方向上分配的误差是对角线上像素的 2 倍，但这种处理方法会产生龟纹。

8.4.5 噪声半色调法

蓝噪声和绿噪声数字半色调算法是上世纪九十年代以后提出来的半色调算法。最早引入半色调技术的是白噪声半色调法，但是由于白噪声的低频成分相当多，而人眼对低频成分相

当敏感，所以该方法产生的半色调图像龟纹多，图像易变形。为了将图像同一灰度中的少数点尽量均匀分散，构造只包含光谱中高频成分的模板，使其具有蓝噪声特性。蓝噪声半色调图像没有按规律性网格排列产生的纹理，但是由于少数点的分散孤立处理，使得蓝噪声模板很容易受到打印机失真和一些处理能力差的设备的影响，绿噪声半色调技术应运而生。与蓝噪声处理白噪声的高频部分不同，绿噪声是白噪声中的中频成分。绿噪声技术可以产生空间分辨率更高的半色调模式，在彩色打印过程中可以消除纹理，同时具有一定的抗打印变形能力，使得算法的健壮性有很大提高^[16]。

1. 蓝噪声半色调法

人眼视觉系统对于低频成分较之高频成分更为敏感，过多的低频成分会使图像看起来杂乱无章，极具颗粒性。没有低频颗粒且不相关的二值图通常具有令人满意的视觉效果，这样的二值图的频谱集中在“蓝色频率”附近，因此它具有蓝噪声特性。理想的具有蓝噪声特性的二值图的径向平均功率谱如图 8-19 所示。

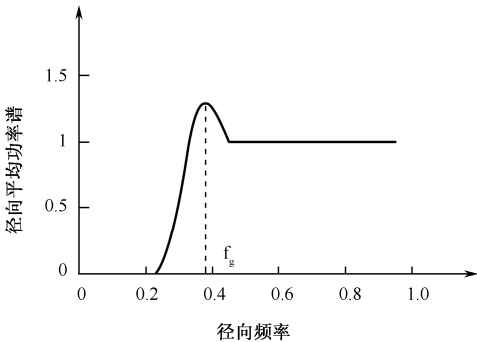


图 8-19 理想的具有蓝噪声特性的二值图的径向平均功率谱

2001 年 V. Ostromoukhov 提出利用蓝噪声特性来改进误差分散法^[17]。对于一幅具有 256 级灰度的连续色调图像，V. Ostromoukhov 在研究了传统误差分散法和蓝噪声半色调技术后得出了以下结论：

- ① 当误差分散的输出图像的傅立叶频谱比较接近蓝噪声频谱时，产生视觉效果很好的半色调图像。一个非周期频谱对称结构不具有低频能量的信号具有蓝噪声特性。
- ② 蓝噪声的误差分散方向如图 8-20 所示。设算法对于灰度级 g_1 ， g_2 已分别产生分散系数 $D^1 = \{d_{10}^1, d_{-11}^1, d_{01}^1\}$ ， $D^2 = \{d_{10}^2, d_{-11}^2, d_{01}^2\}$ ，两组不同的分散系数产生的半色调图像具有明显不同纹理结构。如果对于处于灰度级 g_1 ， g_2 之间的灰度级 g_i ，其分配系数 $D^i = \{d_{10}^i, d_{-11}^i, d_{01}^i\}$ 是根据 D^1 ， D^2 平滑插值产生，那么这两个灰度之间的纹理结构变化曲线比较平滑。最简单的办法是用线性插值算法来产生这组灰度系数。

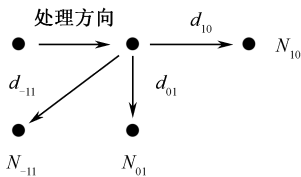


图 8-20 蓝噪声误差分散方向示意图

③ 在某些灰度上容易产生明显的人工纹理，如 1、64、85、127，这些灰度为主灰度。

④ 灰度级 g_i 与 g_{255-i} 的纹理结构是相同的，只是黑点与白点相反。因此，二者的分配系数也相同。

针对以上结论，得到算法的步骤如下：

① 对每一个主灰度值，设法得到一组分配系数 $D^{\text{key}} = \{d_{10}^{\text{key}}, d_{-11}^{\text{key}}, d_{01}^{\text{key}}\}$ 使得输出图像的傅立叶频谱和蓝噪声频谱尽可能接近。这是一个非常经典的最小值问题，一开始设该灰度级误差分散系数为 $\{1/3, 1/3, 1/3\}$ ，然后分别改变各个分量，设法得到输出图像傅立叶频谱和蓝噪声频谱的差值最小，尽量逼近理想蓝噪声频谱。

② 在两个主灰度值间采用线性插值，这样可以产生光滑的纹理结构曲线。如果在某些灰度产生的视觉结构不好，则增加这种不好的灰度到最初的主灰度值上，从第①步重新开始处理。

③ 以 127.5 为对称中心将 0~127 的分配系数扩展： $D^{128} = D^{127}$ ， $D^{129} = D^{126}$ ，……， $D^{255} = D^0$ 。

这种误差分散法只有三个误差分配系数，因此处理速度更快。经过蓝噪声误差分散法处理的半色调图像中点的分布更加均匀，其对应的频谱图像具有更好的对称性，更接近蓝噪声频谱，并且具备与 Floyd-Steinberg 误差分散法相同的简单性。

2. 绿噪声半色调法

尽管蓝噪声误差扩散半色调图案有良好的功率谱特征，但用于调幅和调频复合加网工艺时却并不适用，为此 D. L. Lau 等提出了绿噪声的概念^[18]。绿噪声与蓝噪声一样具有非周期性，有不相关的结构，没有低频颗粒。与蓝噪声不同的是，绿噪声半色调以簇点为单位进行处理，没有蓝噪声的高频部分，类似于可见光的中频部分，所以称为“绿噪声”。

D. L. Lau 等利用 BIPPCCA (Binry-Pattern-Pair-Construction-Algorithm) 算法构建各个灰度级的二值抖动模板^{[18][19]}。然后生成模板矩阵。BIPPCCA 算法首先产生一个全为 0 的 $M \times N$ 矩阵 $G_{M \times N}$ ，矩阵中每个元素都记录下其成为少数点的概率存放在矩阵 $U_{M \times N}$ 中，然后根据 $U_{M \times N}$ 依次把矩阵 $G_{M \times N}$ 中的元素转化成少数点，转化的截止条件是矩阵中黑点所占的比例等于所需的灰度级 g 。在每次反复迭代计算中，概率值最大的元素最先转化成少数点。同时，为了使生成的抖动矩阵具有绿噪声的空间统计特性，每次转化完成后得到新的 $G_{M \times N}$ ，BIPPCCA 会根据现在的少数点集合调整矩阵 $G_{M \times N}$ 每个多数点转化成少数点的概率^[20]。

BIPPCCA 算法在初次转化少数点之后会根据点相关函数 $\bar{R}(r)$ 调整剩下多数点的概率值。 $\bar{R}(r)$ 由理想化的绿噪声点相关函数 $R(r)$ 派生出来，与 $R(r)$ 近似，但更容易实际计算，如图 8-21。

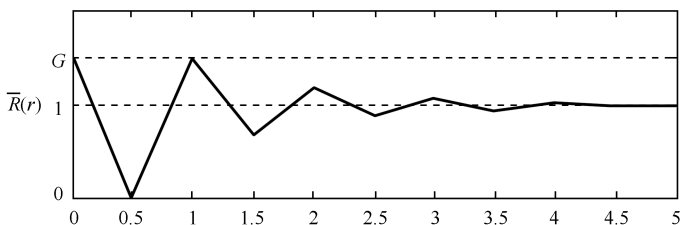


图 8-21 构建绿噪声抖动模板的点相关函数表

从图中看出, 这是一个分段线性函数, 而且在 λ_g 的倍数处有波峰。如果多数点与新增的少数点的距离所对应的 $\bar{R}(r)$ 值大于 1, 多数点转化概率值增加; 反之, 则减少。为了克服 $\bar{R}(r)$ 的不稳定性, BIPPCCS 引入浓度矩阵 CM, CM 增加了在少数点的低浓度区中的多数点转化概率值, 同时减少少数点在高浓度区中多数多的转化概率值。并设计一个低通滤波器 H_{LP} 对每次迭代产生的新模板作循环卷积, 即 $\{H_{LP} \otimes G_{M \times N}\}$, 得到少数点的浓度, 高斯低通滤波器定义如下:

$$H_{LP}(r) = \exp\left(\frac{-r^2}{\lambda_g^2}\right) \quad (8-14)$$

λ_g 越大, H_{LP} 脉冲的宽度越大; 反之, H_{LP} 是一个窄而尖的脉冲。概率的改变量由少数点像素的浓度通过 H_{LP} 后的输出进行映射而得到的浓度矩阵确定, 映射的实现是把 $\{H_{LP} \otimes G_{M \times N}\}$ 的值线性规划到 0 和 1 之间, 即 $\max\{H_{LP} \otimes G_{M \times N}\} \rightarrow 0$; $\min\{H_{LP} \otimes G_{M \times N}\} \rightarrow 1$ 。

其算法实现如下:

- ① 产生一个全为 0 的 $M \times N$ 初始模板矩阵 $G_{M \times N}$ 。
 - ② 构建与矩阵 $G_{M \times N}$ 相对应的概率矩阵 $U_{M \times N}$ 。 $U_{M \times N}$ 中元素初始值随机产生, 介于 0 与 1 之间。
 - ③ 根据 $\{H_{LP} \otimes G_{M \times N}\}$ 的输出值构造浓度函数 CM。
 - ④ 把 CM 矩阵与概率矩阵 $U_{M \times N}$ 相卷积, 确定 $G_{M \times N}$ 中有最高转化概率的多数点像素 $G_{M \times N}(i, j)$, $G_{M \times N}(i, j)$ 的值由 0 变为 1。
 - ⑤ 根据新的 $G_{M \times N}$ 矩阵, 由点相关函数 $\bar{R}(r)$, 产生新的概率矩阵 $U_{M \times N}$ 。
- $$(U_{M \times N})_{\text{new}} = (U_{M \times N})_{\text{old}} \times \bar{R}(r) \quad (8-15)$$
- ⑥ 如果 $G_{M \times N}$ 中少数点的总数与其矩阵中的像素总数之比等于灰度级 g , 退出循环; 否则, 跳转到步骤③。

BIPPCCA 算法时间复杂度较高, 当给定模板矩阵大小 $M \times N$ 和灰度级 g 时, 每执行一次 BIPPCCA 算法, 都要进行 $g \times M \times N$ 次大循环。时间复杂度高达 $O(n^4)$ 。可以通过改变高斯函数矩阵等方法来提高算法的运行效率^[20]。

8.4.6 影响数字半色调的因素

阈值的选取方法、误差滤波器系数的选取以及处理过程中扫描像素路径的选取是数字半色调技术的核心问题, 同时也是影响数字半色调的三大因素^[21]。

1. 像素的处理路径

众所周知, 误差分散法是一种比较流行且效果较好的方法, 其思想是先阈值量化图像像素, 然后将量化的误差分散到相邻的未处理的像素上。该算法最早是由 Floyd-Steinberg 提出的, 其优点是得到的半色调图像质量效果好, 色调丰富, 像素点的分布是各向异性和无规律的。但该算法的不足之处在于采用该方法得到的半色调图像中易在亮光和暗调部位出现龟纹现象以及与处理方向有关的滞后现象。出现这些现象的主要原因是扫描像素的顺序是逐行扫

描的,为此对扫描处理像素的顺序加以改进,使得再现图像尽可能少地存在以上的不良信息。其改进方法主要有以下几种。

(1) “蛇形”扫描方式

文献[22]提到以“蛇形”扫描的方式代替逐行扫描的方式,采用该扫描方式处理后得到的图像质量明显优于前者,结果图像中亮光和暗调部位出现拖尾现象明显少于前者,从而改进了再现图像的质量。

(2) Hilbert 曲线扫描方式

空间填充曲线^[22, 23]是按照一定的顺序连续扫描图像中的每一个像素,并且每个像素仅扫描一次。最流行的空间填充曲线是 Peano-Hilbert 曲线,该空间填充曲线可以用于数字图像半色调处理。该扫描方式是根据处理的图像大小确定处理图像的 Hilbert 曲线。方法是从该曲线的起点出发,沿着一定大小的 Hilbert 曲线逐个量化、处理每一个像素,直到处理结束。Hilbert 曲线的生成原理如图 8-22 所示。

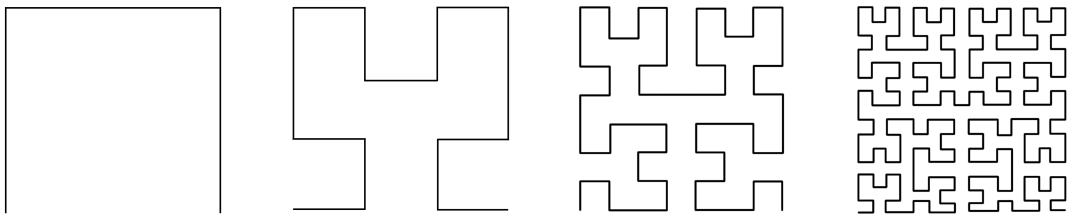


图 8-22 n 取不同值的 Hilbert 曲线

采用该方法处理后得到的半色调图像的质量虽然优于逐行扫描法和蛇形扫描法,再现后图像的效果相对较好,不存在与处理方向有关的滞后现象,规律性的、结构性的纹理相对较少。但是由于沿着该曲线处理像素的过程中,图像边缘处存在许多十字交叉,再现后的图像还存在“龟纹”现象,存在规律性的、结构性的纹理,边缘部位不连续的过渡现象比较严重,处理后的图像边缘不平滑等。为了克服 Hilbert 曲线法处理后图像中的不足,文献^[24]提出了随机曲线填充的扫描方式,采用该方式处理后的再现图像的质量稍优于采用 Hilbert 曲线扫描方式处理后得到的再现图像。但是仍然存在边缘部位不连续的过渡现象,处理后的图像边缘不平滑等不足。

(3) 基于上下文相关的空间填充曲线的扫描方式

基于上下文相关的空间填充曲线是根据图像相邻像素灰度值之间的内在关系来寻求一种扫描图像像素的最佳路径,它反映图像自身的自相关性。其基本思想是:首先定义一个与待处理图像像素个数相等的带权图与一个与之对应的且顶点数是它的四分之一的带权偶对图,其次求其带权偶对图的最小生成树,然后根据最小生成树将非连通的原图连接成连通图,该连通图就是要求的扫描图像像素的最佳路径,最后再对连通图采用深度优先遍历的方法量化、误差分散处理图像的每一个像素即可得到该扫描路径下的半色调图像。沿着该曲线处理像素的过程中,尽可能多地处理图像的某一个区域,图像边缘处存在十字交叉很少。所以,通过该方法处理后的图像质量比前面的任何一种扫描方式处理的图像都好,不仅没有“龟纹”现象,也不存在规律性的、结构性的纹理,处理后的图像边缘平滑,而且处理后图像的自相关性比较高。

三种扫描方式自相关性的比较如图 8-23 所示^[21]，可以看出，采用基于上下文相关的空间填充曲线的扫描方式处理后，图像的自相关性好，因而得到的半色调图像质量高。

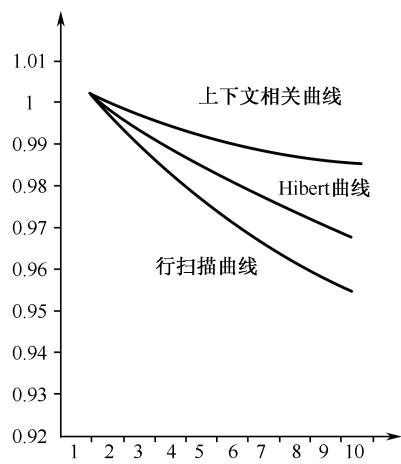


图 8-23 三种扫描方式的自相关性

2. 阈值的选取方法

数字半色调技术处理过程中，对原图像进行量化时，必须要考虑量化阈值的选择问题。量化阈值选则的好坏直接影响到能量误差的传递。量化阈值选择越恰当，能量误差就越小，传递时影响越小。由此可见，量化阈值的选择是影响数字半色调技术的因素之一^[21]。

(1) 中值法

图像的能量通过一系列的数学手段，一般都转化成为一些量化过的整数数据或灰度等级数据。中值法是将图像中的每一个像素的灰度值与设备能再现的灰度等级的一半 P 进行比较，若大于 P 则在最终的半色调图像中被指定为 1，反之则被指定为 0。

其数学表达式为

$$B(x_{m,n}) = \begin{cases} 1, & f(x_{m,n}) > P \\ 0, & f(x_{m,n}) \leq P \end{cases} \tag{8-16}$$

(2) 平均值法

中值方法简单易用，但是并不能很好地反映图像的特征信息，只反映设备再现图像的灰度范围，对于亮调图像、暗调图像都不能反映图像的细节信息，如图像边缘、图像灰度等。在数字半色调技术中，应该选取合适的阈值来反映图像的细节信息。

针对再现图像的特征，选取不同的阈值，其中平均值法是一种最简单的方法。其数学表达式为：

$$V(x_{m,n}) = \begin{cases} 1, & f(x_{m,n}) > P_{m,n} \\ 0, & f(x_{m,n}) \leq P_{m,n} \end{cases} \tag{8-17}$$

其中，邻域的平均值 $P_{m,n}$ 为

$$P_{m,n} = \frac{1}{9} \sum_{\substack{K=-1,+1 \\ L=-1,+1}} f_{m+K,n+L} \tag{8-18}$$

在再现图像的过程中,为了控制再现图像的对比度,加强图像的锐化性能,对式(8-17)、(8-18)的平均阈值法进行了改进,改进的数学表达式为

$$V(x_{m,n}) = \begin{cases} 1, & f(x_{m,n}) > \phi_{m,n} \\ 0, & f(x_{m,n}) \leq \phi_{m,n} \end{cases} \quad (8-19)$$

其中, $\phi_{m,n}$ 为

$$\phi_{m,n} = \gamma + P_{m,n} \times \left(1 - \frac{2 \times \gamma}{f_{\max}}\right) \quad (8-20)$$

f_{\max} 为原图像像素的最大灰度值。参数 γ 决定了最终二值图像的对比度,且与图像像素的噪音统计有关。采用此方法进行半色调处理时, γ 值的恰当选取非常重要。当参数 γ 的绝对值增大时,图像对比度逐渐增大,背景噪声被抑制,景物突出。但是,当参数 γ 的绝对值增大到一定程度时,背景噪声虽进一步抑制,但景物的细节也有一些丢失。

(3) 动态阈值法

在数字半色调技术中,为了充分再现原图像的边缘细节信息,阈值的选取可采用动态阈值法来实现。所谓动态阈值法是指随着处理图像像素的不同,其量化阈值是不同的。实现过程如下。

① 图像的当前像素的灰度值 $f_{m,n}$ 与两个固定的阈值进行比较:

$$V(x_{m,n}) = \begin{cases} 1, & f(x_{m,n}) > A_{\max} \\ 0, & f(x_{m,n}) < A_{\min} \end{cases} \quad (8-21)$$

其中, A_{\max} 、 A_{\min} 是选取的两个阈值。若上式两个条件都不满足,则采用第②步计算相应的阈值。

② 首先使用一个低通滤波器计算邻域平均值 Ave, 其低通滤波器模型为

$$h_1(m,n) = \frac{1}{16} \begin{vmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{vmatrix} \quad (8-22)$$

则邻域平均值 Ave 为

$$\text{Ave} = [4f_{m,n} + 2f_{m,n-1} + 2f_{m,n+1} + 2f_{m-1,n} + 2f_{m+1,n} + f_{m-1,n-1} + f_{m-1,n+1} + f_{m+1,n-1} + f_{m+1,n+1}] / 16 \quad (8-23)$$

其次,根据下式计算一个边缘系数:

$$D_e = \frac{f_{m,n} - f_{m-1,n}}{2} + \frac{f_{m,n} - f_{m,n-1}}{2}$$

最后得到的阈值为

$$P_{m,n} = m \times D_e + \alpha \times \text{Ave} \quad (8-24)$$

其中, m 为边缘敏感系数。在动态阈值中,可以通过调节 α 和 m 的值提高半色调图像的质量。

(4) 极值点阈值法

一般情况下,图像分两个大的层次:图像背景和目标两部分。而图像的背景和目标有明显的区别,故可以根据图像的直方图进行阈值的选取。最简单的直方图阈值选取法是极值点阈值法,该方法是将图像直方图的包络看成一条曲线,则选取直方图的谷可借助求曲线极小

值的方法。设 $h(z)$ 代表图像的直方图，那么极小值点应满足

$$\frac{\delta h(z)}{\delta z} = 0 \text{ 或 } \frac{\delta^2 h(z)}{\delta z^2} = 0$$

则满足上式极小值点对应的灰度值可作为半色调处理时的阈值。

(5) 最优阈值法

虽然图像在一般情况下可以分为背景和目标两部分，但是有时图像的目标和背景的灰度值有部分交错，对于满足上式的极小值点有多个，那么选取哪一个极小值点作为相应的阈值进行半色调处理才能使所得图像质量最佳呢？最优阈值选取法是一种常用的方法，该方法的基本思想是：待处理图像的直方图可以看成灰度值概率密度函数 $p(z)$ 的一个近似值，设该图像灰度值的混合概率密度函数 $p(z)$ 由下式计算：

$$p(z) = p_1 p_1(z) + p_2 p_2(z) = \frac{P_1}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{(z-\mu_1)^2}{2\sigma_1^2}\right] + \frac{P_2}{\sqrt{2\pi}\sigma_2} \exp\left[-\frac{(z-\mu_2)^2}{2\sigma_2^2}\right] \quad (8-25)$$

其中， μ_1 、 μ_2 分别是背景和目标区域的平均灰度值， σ_1 、 σ_2 分别是背景和目标区域的均方差， P_1 、 P_2 分别是背景和目标区域灰度值的先验概率，且 $P_1 + P_2 = 1$ 。该混合概率密度函数 $p(z)$ 的参数可根据最小均方误差方法借助直方图得到。若 $\sigma_1 = \sigma_2 = 0$ ，则满足上式的最优阈值只有一个：

$$T_{\text{optimal}} = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \ln\left(\frac{P_2}{P_1}\right) \quad (8-26)$$

若 $P_1 = P_2$ ，则最优阈值就是两个区域中平均灰度值的中值。

(6) 依赖坐标的阈值选取方法

依赖坐标的阈值选取方法的基本思想是先将图像分解成一系列子图像，这些子图像可以相互重叠也可以只相邻。然后可用任意一种固定阈值法对每个子图像计算一个阈值，最后，通过对这些子图像的阈值进行插值运算即可得到图像中每个像素进行处理时的阈值。

3. 误差滤波器的设置

数字半色调技术处理过程中，对原图像进行量化后，必须要考虑量化误差的分散问题。在误差分散过程中，选取不同的误差滤波器系数，得到半色调图像的质量不同。误差滤波器系数选择的好坏直接影响到能量误差的传递。误差滤波器系数的设置也是影响数字半色调技术的因素之一。

(1) 时序滤波器法

时序滤波器法是将能量误差按图像处理的扫描顺序依次传递到相邻区域的一种滤波器设计方法。由于人眼是低通滤波器，图像的能量集中在低频区域，同时图像的重要区域在其低频阶段，因而必须把能量传递的方式设计成一个低通滤波器。考虑到人眼的观察点同图像像素点的位置关系，能量应该向近点传递得多，向相隔较远点传递得少。经典的低通时序滤波器有 Floyd-Steinberg 滤波器、Stucki 滤波器、Sierra 滤波器等。

在数字半色调处理的过程中，采用不同的滤波器系数，得到半色调图像的质量不同，相对而言采用其中后两种得到的图像质量较好，但是它们的计算量比较大。

(2) 非时序滤波器法

在数字半色调技术处理的过程中,由于图像像素点的相互影响不具有时序性,因而采用时序滤波器法并不能很好地反映图像像素之间的关系,反而使得处理后得到的半色调图像出现许多规律性的、结构性的纹理,在高光和暗调部位出现龟纹现象以及与处理方向有关的滞后现象等。所以,在半色调处理的过程中,提出了采用非时序滤波器法进行误差能量之间的传递,使得处理后的图像出现上述不足较少,从而大大改善再现图像的质量。由于人眼对水平垂直方向误差能量的敏感程度与对角线方向误差能量的敏感程度不同,所以非时序滤波器系数模板中,水平垂直方向误差分配系数大于对角线方向误差分配系数,从而能够反映人眼的视觉特性。具体的非时序滤波器系数模板如图 8-24 所示。

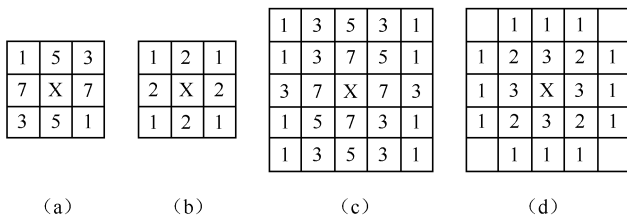


图 8-24 非时序滤波器系数模板

通过实验可知,采用图 8-24 (b) 的模板系数比采用图 8-24 (a) 的模板系数进行误差分散得到的半色调图像质量高^[21]。

总而言之,图像再现过程中,扫描路径、阈值和滤波器误差系数的选取是数字半色调技术的核心问题,是影响该技术的三个关键因素,直接影响到再现图像的质量。

8.5 半色调数字水印技术

目前国内外已经有学者、公司等开始了关于半色调图像水印技术的研究,就研究成果而言,大都局限在实验阶段,还没有成熟的商品化软件推出。下面对现有的半色调图像的数字水印算法进行分类介绍。

8.5.1 半色调水印技术的基本方法

1. 直接嵌入法

直接嵌入法是指在半色调化后的二值图像中直接嵌入水印信息,其处理过程如图 8-25 所示,该类水印嵌入算法的复杂度普遍较低,易实现。

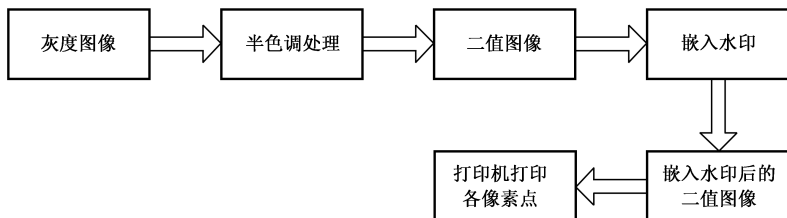


图 8-25 直接嵌入法

M. S. Fu 和 O. C. Au 提出了直接在误差分散半色调图中嵌入水印的三种算法^[4], 即 DHST (Data Hiding Self Toggling)、DHPT(Data Hiding Pair Toggling)、DHSPT(Data Hiding Smart Pair Toggling)。该类算法中选取的水印信息都是一幅二值图像, 水印图像的大小远小于载体图像(原半色调图像)的大小。例如原图为 256×256 的二值图像, 水印图像大小可选为 32×32 。

DHST 算法是在半色调图像中随机选择 N (水印图像的像素总数) 个水印嵌入位置, 并强制该位置上的像素值等于水印值。具体嵌入过程如下。

- ① 用伪随机数产生器产生 N 个随机位置, 作为水印的嵌入位置。
- ② 在每个随机位置上, 如果该位置上的像素值与要嵌入的水印像素值相同, 则不做处理; 否则, 进行修改, 使该位置的像素值与水印值相同, 从而实现水印的嵌入。

提取水印时, 用伪随机数产生器产生相同的 N 个随机位置, 从这些位置提取相应的数据即可以重现嵌入的水印。该方法算法复杂度很低, 容易实现。但是由于是强行改变像素值, 没有考虑嵌入水印位置与周围像素点的联系, 导致引入了很多椒盐噪声, 嵌入水印后的图像质量较差。

DHPT 算法在 DHSP 算法的基础上做了改进, 尽量减少嵌入水印给图像带来的影响, 即减少图像局部平均亮度的变化。具体步骤如下。

- ① 用伪随机数产生器产生 N 个随机位置。
- ② 在每个随机位置上, 如果该位置上的像素值与要嵌入的水印像素值相同, 则不做处理; 否则, 进行修改, 使该位置的像素 (“主” 像素) 值与水印值相同, 同时在它的 3×3 邻域内选择一个互补的像素 (“从” 像素) 进行修改。通过同时修改互补的一对 “主”、“从” 像素点来嵌入水印, 从而减小图像局部平均亮度的变化。

提取水印时只考虑主像素点即可, 方法与 DHSP 算法的水印提取方法相同。

由于 DHPT 算法中的 “从” 像素的位置是随机选择的, 因此也会产生 “椒盐” 噪声, 鉴于这种情况, DHSPT 算法给出了一种选择最优 “从” 像素的方法。这里考虑 (m,n) 位置处的像素点和它 3×3 邻域内的像素。将这个 y 邻域内的像素值标记为 $[x_1, x_2, x_3, x_4, x_0, x_5, x_6, x_7, x_8]$, x_0 作为 (m,n) 处的像素值。该算法设置了一个 $\text{con}(m,n)$ 函数, 如下所示:

$$\text{con}(m,n) = \sum_{i=1}^8 w(i)f(x_0, x_i) \tag{8-27}$$

其中, $f(x,y) = \begin{cases} 1, & x = y \\ 0, & x \neq y \end{cases}$

当 $i=1,3,6,8$ 时, $w(i)=1$; 当 $i=2,4,5,7$ 时, $w(i)=2$ 。这是因为中心点上下左右的像素点离它更近, 所以赋以更大的权值。

$\text{con}(m,n)$ 表示的是主像素与其邻域内像素值相同的点的关联程度, 通过计算主像素点 3×3 邻域中各个点的 $\text{con}(m,n)$ 值, 最终选取从像素改变后 $\text{con}(m,n)$ 值最小的点为最优从像素点。

利用以上三种算法嵌入水印后的图像或多或少都存在 “椒盐” 噪声, 为改善其视觉效果 Ping-Sung Liao 等人提出了一种无损的半色调水印技术。

该方法的主要步骤是: 首先对半色调图进行无重叠分块, 在这些分块矩阵中通常会出现很多相同的矩阵, 将出现频率高的矩阵块作为 “候选模板”; 然后构造与候选模板对应的 “配对模板”, 配对模板要求满足与它对应的候选模板的汉明距离为 1, 以此保证这对模板之间的差异最小, 但是选取的这个配对模板又不同于其他的候选模板和配对模板。通过这种选择方

法,能够得到合法配对模板的区域,其作为嵌入水印的区域;最后用候选模板表示 1 (0)、配对模板表示 0 (1),利用模板替换法嵌入水印。

提取水印的方法是在事先保存的水印嵌入位置中提取到模板,将它与原有的模板库相比较,来得到嵌入的水印值。

该算法适用于各种半色调图,可以通过模板替换无损失地恢复原始半色调图,而且视觉效果也比较好。但是由于该算法处理任何图像时都需要重新统计候选模板,并计算相应的配对模板,所以算法的复杂度较高。并且此算法并没有在打印扫描环境中进行实验。经验证,该算法的抗打印扫描的攻击能力较弱,难以正确提取出水印。

2. 间接嵌入法

典型的间接嵌入法是根据二值水印,选择满足相容性和互异性的有序抖动模板或误差分散核,再利用半色调过程嵌入水印的。这种方法的优点是不增加任何额外计算,但是构造抖动模板或误差分散核比较困难。根据不同抖动模板所产生的半色调图像具有的不同统计特性来检测水印,这种方法的水印检测的正确率较高,但图像的效果一般,而误差分散的因果性会导致半色调图像的统计特性发生改变,其水印检测的正确率稍差。

Soo-Chang P 等人针对有序抖动图像提出了 PSMOD (Paired Sub-image Matching Ordered Dithering) 算法^[8],该算法的主要步骤如下。

① 用大小相间的有序抖动模板进行半色调处理。

② 对半色调图进行位交错 (bit-interleaving) 预处理,得到明暗相间的子图;通过调整明暗子图对的顺序来嵌入二值水印。

为了增大水印嵌入量,可以采用增大有序抖动模板,或对位交错子图再进子图交错处理 (sub-image-interleaving) 的方法来生成更多明、暗相间的子图,从而增大水印嵌入量。实验表明这种方法的错误率更低,效果更好。这种算法利用了区域统计特性,对涂改、剪切有很好的鲁棒性,而且水印嵌入量比较灵活,复杂度较低,但是由于该算法仅适用于有序抖动图像,而有序抖动方法得到的半色调图像的效果有不足之处,所以它的应用范围受到了限制。

利用误差分散法的半色调过程嵌入水印也是一种方法,王向阳等人提出了一种基于神经元的误差分散法^[25]。Adaline 神经元模型的描述如下:设神经元输入信号矢量为 $X = (x_0, x_1, \dots, x_n)^T$, 权矢量为 $W = (w_0, w_1, \dots, w_n)^T$, 净输入 $\text{net} = X^T \cdot W = X \cdot W^T$, 相应的二值输出及转移函数为

$$y = \frac{1}{2}(f(\text{net} - T) + 1), f(t) = \text{sqn}(t) = \begin{cases} 1, & t \geq 0 \\ -1, & t < 0 \end{cases} \quad (8-28)$$

其中, T 为神经元阈值,当 $\text{net} - T \geq 0$ 时,神经元才被激活。

这种数字水印方法的基本思想是将误差分散所采用的 Steinberg 误差分散核看成 Adaline 神经元,只是 Steinberg 滤波器不具备 Adaline 神经元的训练功能。以原始灰度图像 (i, j) 处像素的邻域内各像素所产生误差作为输入信号矢量,以 Floyd-Steinberg 滤波器系数为权矢量,将误差分散过程重新定义为

$$a(i, j) = W^T \cdot X = \begin{bmatrix} 1 & 5 & 3 & 7 \\ 16 & 16 & 16 & 16 \end{bmatrix} \cdot \begin{bmatrix} e(i-1, j-1) \\ e(i-1, j) \\ e(i-1, j+1) \\ e(i, j-1) \end{bmatrix} \quad (8-29)$$

$$\begin{aligned} y(i, j) &= \frac{1}{2}(f(\text{net} - T) + 1) \\ &= \frac{1}{2}\{\text{sqn}[g(i, j) + a(i, j) - T] + 1\} \\ &= \frac{1}{2}\{\text{sqn}[f(i, j) + -T] + 1\} \end{aligned} \tag{8-30}$$

水印的嵌入过程是：如果选取的水印位置的像素点经过半色调处理后的值与水印信息相同，则不变；否则将水印信息位 $m(k)$ 作为理想输出，利用 Adaline 神经元的训练方法对误差分散核的系数进行训练，直到得到的误差分散系数可将水印位置的像素点训练成指定的值（0 或 1），然后利用得到的新系数继续对图像进行半色调处理。

提取水印的方法是利用伪随机发生器产生相同的随机位置，在该位置上直接提取像素点组成水印信息即可。

该方法提出了一种很好的在半色调过程中嵌入水印的思想，利用误差分散法将水印产生的误差分散出去，能够很好地克服椒盐噪声的问题。但是由于此方法使用神经元训练出的误差分散核的系数有正有负，在使用新系数进行半色调处理的过程中，图像的纹理性明显，因此需要进一步改进^[1]。

8.5.2 核转换误差分散水印算法

在文献[26]中，S. C. Pei 和 J. M. Guo 提出了一种核转换误差分散（Kernels-Alternated Error Diffusion, KAEDF）水印算法，通过在半色调变换过程中改变核而实现不可见水印的嵌入。它的算法原理是：选取两个相容性较好的误差分散核，根据水印信息交替使用误差分散核对原图进行半色调处理，最终获得嵌入水印的半色调图像，该算法框图如图 8-26 所示。

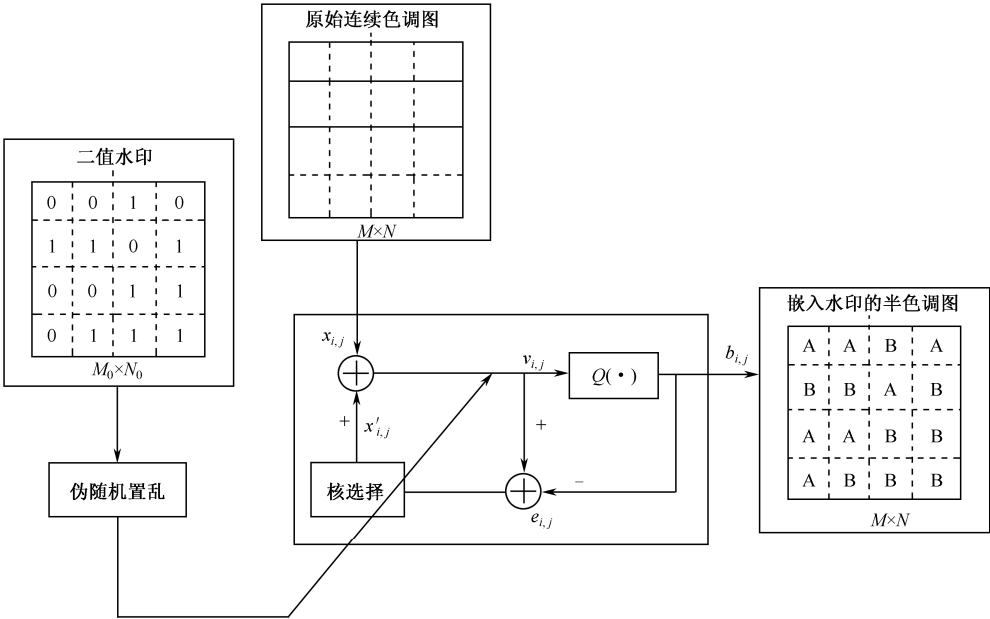


图 8-26 KAEDF 算法流程图

KAEDF 算法流程是：设定原始数字图像大小为 $M \times N$ ，二值水印大小为 $M_0 \times N_0$ ，将原始图像分割成大小为 $\frac{M}{M_0} \times \frac{N}{N_0}$ 的小块，然后根据水印信息选择不同的误差分散核对每块进行半色调处理，如水印信息为 0 采用 A 核，水印信息为 1 采用 B 核，重复这种处理过程，最后得到嵌入水印的半色调图像。KAEDF 算法主要是选择两个相容的误差分散核，通过实验得到 Jarvis 和 Stucki 两个误差分散核具有最大相容性。

KAEDF 算法实现简单且具有一定的鲁棒性，但是它所生成的含水印半色调图像仍然具有一般误差分散算法固有的边缘锐化和噪声成形失真。而在水印检测时，其核判决阈值要由原始连续色调图像产生，因此 KAEDF 算法本质上是非盲的。

文献[27]中，T. D. Kite 等人将量化器模型化为信号的增益与一个附加噪声的叠加（图 8-27），量化器的输出由此也分成信号部分 $y_s(i, j)$ 和噪声部分 $y_n(i, j)$ 。

$$Y(z) = \text{STF} \cdot X(z) + \text{NTF} \cdot N(z) \quad (8-31)$$

其中，STF 为信号传输函数，NTF 为噪声传输函数。

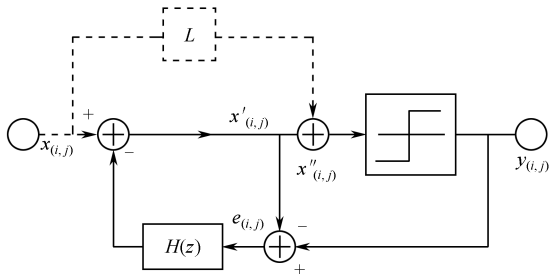


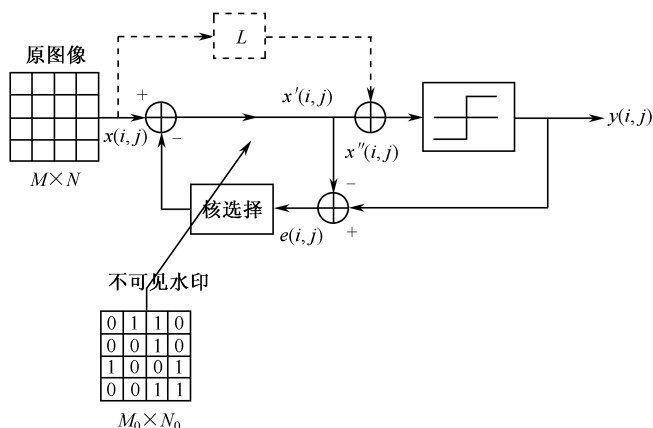
图 8-27 改进的误差分散算法

在图 8-27 中，除去虚线部分为标准的误差分散算法的框图，其中 $x(i, j)$ 代表输入连续像素值， $y(i, j)$ 代表输出二值像素值， $-0.5 \leq x(i, j) \leq 0.5$ ， $y(i, j) \in \{-0.5, 0.5\}$ ； $e(i, j)$ 是量化误差， $H(z)$ 是误差滤波器或核。

实验仿真结果表明：标准误差分散算法中常用的几种核所对应的 STF 在高频段的增益都远大于 1，这可以解释为标准误差分散算法所固有的边缘锐化失真。同时，由于锐化失真，不可避免地带来了噪声成形失真，即量化误差图像中具有原图像的明显轮廓^[28]。T. D. Kite 等通过在量化器的输入端上增加一个乘法因子 L （图 8-27 虚线部分），来削弱 STF 的增益，使整体信号增益为 1，从而显著改善了所生成半色调图像的质量。

在文献[27]工作的基础上，文献[28]首先发展了一种面向不可见水印的改进的 KAEDF 算法（MKAEDF）（图 8-28），通过采用误差分散算法的量化器模型和阈值调制技术，显著提高了半色调水印图像的视觉质量；与此相应，半色调变换中不同核所对应的 DFT 域的频谱分布特征更好地被保留在所生成的半色调水印图像中，故可由反半色调变换直接生成的连续色调图像计算不同核的判决阈值，从而实现了半色调水印图像的盲检测。

为了克服原 KAEDF 算法在水印图像质量方面的不足，文献[29]在水印嵌入时采用改进的误差分散算法代替原来的标准误差分散算法。由于 Jarvis 核和 Stucki 核的兼容性，算法在嵌入时分别使用 Jarvis 和 Stucki 核表示水印信息 0 和 1。



对于可见水印, 文献[29]给出了一种基于误差分散的半色调可见水印算法 JHVW; 通过融合 MKAEDF 和 JHVW, 文献[28]进一步发展了一种改进的多用途 KAEDF 半色调水印算法, 可以同时实现半色调图像中可见水印和不可见水印的嵌入。

通过融合可嵌入可见水印的 JHVW 算法, MKAEDF 算法被进一步发展成多用途半色调图像水印算法。以上算法融合的主要依据是:

- ① 两种水印嵌入算法都以误差分散为基础;
- ② 可见水印的嵌入并不会明显改变各分块中核的频谱分布, 因此对不可见水印的鲁棒性影响有限。

MKAEDFM 算法包含 MKAEDF, 如果只嵌入不可见水印, 则 MKAEDFM 退化为 MKAEDF。图 8-29 给出了 MKAEDFM 算法的框图。

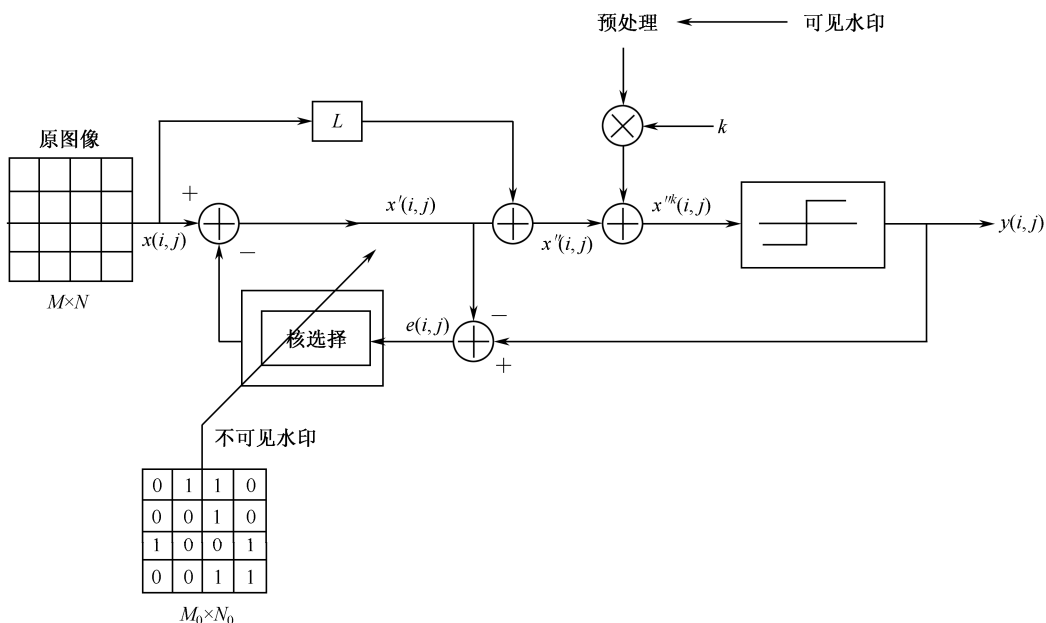


图 8-29 改进的多用途核转换误差分散半色调水印算法框图

其中 (i, j) 为当前输入灰度图像像素点, $x'(i, j)$ 为更新的灰度像素点的输出, $y(i, j)$ 为当前的半色调图像像素点输出, $e(i, j)$ 是量化误差, $x''(i, j)$ 是经过阈值调制后的灰度像素点输出, $x'''(i, j)$ 为嵌入可见水印后量化器的输入, $h(i, j)^{\wedge}$ 为误差分散核, $w(i, j)$ 为可见水印图像, k 表示可见水印的嵌入强度。

为实现 MKAEDFM 中对可见水印的嵌入, 需要对可见水印进行如下预处理操作:

① 考虑到 MKAEDF 量化器的判决门限为 0.5, 需要事先对待嵌入可见水印(灰度图)进行归一化处理;

② 由于 JHVV 是基于像素操作的, 要求可见水印和宿主图像大小一致, 因此需要依据宿主图像对水印图像进行匹配操作。

多用途水印算法 MKAEDFM 的设计目标是在可以清晰地显示可见水印的同时保持对不可见水印的高解码率, 其可调设计参数为可见水印嵌入强度 $k, k \in [0, 1]$ 。为同时满足可见水印清晰度和不可见水印解码率, k 取值在 0.5~0.6 之间是比较合适的^[28]。

8.5.3 半色调水印存在问题 and 研究前景

由于在半色调图像中嵌入水印的难度较大, 因此尽管有越来越多的人投身到半色调水印技术的研究, 并取得了一些成绩, 但与其他多媒体数字水印技术相比, 许多问题的研究尚处于初级阶段, 有待进一步研究解决^[30]。

① 直接嵌入法和大部分间接嵌入法没有很好地利用人眼视觉系统特性。在没有连续色调图像的情况下, 直接嵌入法是唯一的选择, 但现有的直接嵌入法没有很好地利用视觉系统的特性, 且嵌入水印后的图像质量较差。因此要充分利用人眼视觉特性, 在满足不可见性的要求下合理分配水印。另外, 半色调图像大部分基于纸质载体, 由于基于 HVS 的间接嵌入法不可见性较好, 且水印嵌入量较大, 因此该方法应是未来发展的主流。

② 基于变换域的半色调水印尚未得到充分研究。在变换域中嵌入水印信号, 其能量可分布到空域的所有像素, 也有利于保证不可见性和鲁棒性, 虽然目前已有成功的尝试, 但由于复杂度较高, 而且嵌入的水印为伪随机序列, 因此如何利用变换域方法在半色调图像中嵌入有意义的水印, 仍是一个重要的研究方向。

③ 彩色半色调图像水印的研究较少。由于彩色图像提供了比灰度图像更为丰富的信息, 因此彩色图像的处理正受到人们越来越多的关注。由于嵌入水印后的图像效果较差, 因此考虑彩色图像各通道之间的相关性的半色调水印方法仍须进一步研究。

④ PS 失真校正技术、抵抗 PS 过程的半色调水印算法研究还不够。通常打印后的图像, 都有轻微旋转, 其尺寸通常会变大, 由于受打印机“点增益”的影响, 使打印图像比理想图像更黑; 另外经打印、扫描后的半色调图像, 其形状、分辨率都与原图不同, 这将导致水印检测正确率降低, 甚至失败。目前, 大多数算法对打印、扫描的分辨率有特殊要求, 而且检测水印之前需首先进行预处理校正或手工校正, 校正虽然能部分克服 PS 过程的攻击, 但校正后的图像往往与未受攻击的半色调图像相差很大, 鲁棒性较差, 因此须引入纠错码等技术来增强鲁棒性。针对这种现状, 对打印、扫描产生的各种失真进行校正的技术还须进一步研究, 对打印、扫描过程进行研究, 提出能抵抗打印、扫描攻击的半色调水印算法将更有意义。

⑤ 结合半色调图像压缩的半色调图像水印算法尚未研究。与原始图像相比,半色调图像的数据量已大大降低,但是大尺寸的半色调图(如航拍地图)的数据量仍然可观,当传输到印刷/绘图机时,由于会引起数据延迟,因此在半色调图像中嵌入水印时,有必要考虑半色调图像的压缩,其研究具有很重要的实用价值。

参考文献

- [1] 张喻. 抵抗硬拷贝攻击的半色调图像数字水印算法研究. 西安电子科技大学硕士学位论文, 2010.
- [2] Ching-Yung Lin, Shih-Fu Chang. Distortion Modeling and Invariant Extraction for Digital Image Print-and-Scan Process. ISMIP99, Taipei, Taiwan, 1999.
- [3] J. P. Allebach, D. Kacher. Joint halftoning and watermarking. In Proceedings of the IEEE International Conference on Image Processing. Vancouver, BC. 2000, 487-489.
- [4] M. S. Fu 和 O. C. Au. Data hiding watermarking for halftone images. IEEE Transactions on Image Processing. 2002, 11(4):477-484.
- [5] Xu chao-yong. Digital halftoning image watermarking based on conditional probability. Kaohsiung: National Kaohsiung First University of Science and Technology. 2002.
- [6] 张冠男, 王树勋, 温泉. 一种基于随机误差分散技术的半色调水印方法. 吉林大学学报(工学版). 2004, 34(4):639-643.
- [7] 牛少彰, 钮心忻, 扬义先, 胡文庆. 半色调图像中数据隐藏算法. 电子学报. 2004, 32(7): 1180-1183.
- [8] Soo-Chang P. Novel robust watermarking technique in dithering halftone images. IEEE Signal Processing Letter. 2005, 12(4):333-336.
- [9] M. Wu, B. Liu. Data Hiding in Binary Image for Authentication and Annotation. IEEE Transactions on Multimedia. 2004, 6(4):528-538.
- [10] Yang huijuan, Kot, Alex C.. Pattern-based data hiding for binary image authentication by connectivity-preserving. IEEE Transactions on Multimedia. 2007, 9(3):475-486.
- [11] 徐佳. 抗打印扫描数字水印算法在印刷防伪中的应用. 北京印刷学院硕士学位论文, 2012.
- [12] L. L. Daniel. Modern digital halftone. Ph. D. dissertation. Newark, Delaware, USA: University of Delaware, 1999.
- [13] B. E. Bayer. An optimum method for two-level rendition of continuous-tone pictures [C]. In: Proc IEEE International Conference on Communication, 1973: 2611-2615.
- [14] R. W. Floyd, L. Steinberg. Adaptive algorithm for spatial grey scale [C]. In: Proc SID Int Symp Teach Papers, 1975: 36-37.
- [15] D. E. Knuth. Digital halftones by dot diffusion, ACM. Tr. On Graphics, 1987, 1(6):245-273.
- [16] 周正林. 基于噪声特性的数字半调技术的研究. 西安电子科技大学硕士学位论文, 2005.
- [17] V. Ostromoukhov. A Simple and Efficient Error-Diffusion Algorithm [C]. Computer Graphics Proceedings, SIGGRAPH 2001, pp:567-572.

- [18] D. L. Lua, G. R. Arce, N. C. Gallahger. Green noise digital halftoning. Poreeedings of IEEE. 1998.12, 86(12): 2424-2444.
- [19] B. C. Cooper , D. L. Lau. An Evaluation of Green-noise Mask for Electrophotographic. IS&T's and SPIE Electronic Imaging Expro, 2000,22-28.
- [20] 王成林. 半色调混合加网技术研究, 江南大学硕士学位论文, 2008
- [21] 史琳. 数字半色调技术研究. 西安电子科技大学硕士学位论文, 2007.
- [22] L.Velho, J.M. Gomes. Digital halftoning with space filling curves.Computer Graphics. 1991,25(7):81-90.
- [23] I. H. Witten, R. M. Neal Using peano curves for bilevel display of continuoustone images. IEEE Computer Graphics and Application.1982,2(5):47-52.
- [24] T. Asano. Digital halftoning algorithm based on random space-filling curve. IEICE Trans.Fundamentals.1999,E82-A(03):553-556
- [25] 王向阳, 邬俊. 一种用于半色调图像的数字水印嵌入方法. 小型微型计算机系统, 2007, 28(3): 537-541.
- [26] S. C. Pei, J. M. Guo. Hybrid pixel-based data hiding and block-based watermarking for error-diffused halftone images. IEEE Trans Circuits and System, 2003,13(8):867-884.
- [27] T. D. Kite, B. L. Evans, A. C. Bovik. Modeling and quality assessment of halftoning by error diffusion. IEEE Trans on Image Proc. 2000,9(5): 909-922.
- [28] 倪江群, 等. 改进的多用途KAEDF半色调图像水印算法. 通信学报, 2008, 29(10): 30-36.
- [29] H. Luo, J. S. Pan, Z. M. Lu, et al. Joint halftoning and visible watermarking. IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing. Pasadena, USA, 2006, 109-112.
- [30] 郑海红等. 鲁棒的半色调图像水印综述. 中国图象图形学报, 2007, 12(25): 782-788.

抗几何攻击的数字水印算法

鲁棒图像水印算法经过多年的发展，在研究与应用方面取得了较大的进展，但是早期的图像水印算法主要针对常规的信号处理攻击，如 JPEG 压缩、噪声攻击、图像增强、图像滤波等。但是在实际的应用中往往需要对图像进行旋转、缩放、平移、剪切等操作，这些操作被称为几何攻击，例如，图像的打印扫描过程可能引起图像的轻微旋转和缩放。经过几何攻击后，水印检测会失去同步，即使含水印载体遭受轻微的几何变形，如小角度的旋转、小尺度的缩放、小位置的平移和轻微的几何扭曲，虽然水印信息没有受到破坏，但也无法正确地检测和提取出来。基于传统的变换域和失真校正的方法不能很好地解决几何攻击导致的同步信息丢失问题，因此，抗几何攻击的数字水印算法是数字水印研究领域的一个难点。

近年来，随着多媒体数字版权技术的不断发展以及数字权益保护技术对数字水印鲁棒性要求的不断提高，抗几何攻击的数字水印算法成为了研究者普遍关注的重要的研究课题，国内外研究者从不同的角度和方向出发提出了各种抗几何攻击的水印算法，如基于几何校正的方法和基于不变域的水印嵌入。但这些方法和技术，都只考虑如何在空域或变换系数中嵌入水印信息，而忽略了图像本身的特征。这类方法称为第一代水印技术。这些方法尽管能够在某种程度上抵抗几何攻击，但是依然不能同时抵抗裁剪、纵横比改变等攻击，并且计算开销较大。

图像的特征往往代表图像最本质的特性，而且这些特性不管是在受到常规的信号处理攻击还是几何攻击时都保持相对稳定的状态。因此，如果能将水印信息与这些图像特征相联系，就能保证水印对于这些攻击的不变性。基于图像特征的水印算法正是利用这一思想来实现水印嵌入和水印提取的同步性的。

基于图像特征的算法属于第二数字水印技术，最早由 Kutter 于 1999 年提出^[1]。之后许多学者在这一框架下相继提出了各种不同的算法。在图像的众多特征当中，特征点由于其提取简单、稳定性好等优点，在抗几何攻击数字水印算法中得到了广泛的应用。

9.1 几何攻击

数字水印系统中的几何攻击可以分为局部几何攻击和全局几何攻击。全局几何攻击主要包括旋转、缩放、平移、改变长宽比、拉伸扭曲、翻转等仿射变换以及投射变换等。局部几何攻击包括图像的随机扭曲变形、剪裁、行/列删除等。

9.1.1 全局几何攻击

针对含水印图像的旋转、缩放、平移三种变换又称 RST (Rotation, Scaling, Translation) 攻击, 也是图像水印系统中最常见的全局几何攻击。这三种攻击在实际中很容易实现, 如常用的图像处理软件 Photoshop、ACDSee 等就可以实现对图像的上述操作, 且变换以后的图像在视觉质量上不会产生明显的变化。我们主要对这三种攻击进行介绍。

图像的旋转、缩放、平移攻击可以分别用下面的模型表示^[2]。

1. 旋转 (Rotation)

数字图像的旋转操作实际上是将图像内的每一个像素沿圆形路径进行移动。设图像旋转前的像素坐标为 (x, y) , 旋转角度为 θ , 旋转后的像素坐标为 (x', y') , 则有以下关系:

$$\begin{cases} x' = x \cos \theta + y \sin \theta \\ y' = -x \sin \theta + y \cos \theta \end{cases} \quad (9-1)$$

图 9-1 为对 Lena 图像进行旋转操作的例子。一般情况下, 对一幅图像旋转以后 (除去旋转 90 度的整数倍), 由于数字图像的离散性, 图像的尺度会变大 (通常是在原始图像信息的周围补 0 或补 1), 如图 9-1 (b) 所示。有时候为了使旋转以后的图像与原始图像具有相同的大小, 将旋转后图像的一部分边缘信息剪切掉, 如图 9-1 (c) 所示^[3]。



(a) 原始图像



(b) 逆时针旋转 10 度



(c) 顺时针旋转 10 度加剪切

图 9-1 图像旋转

2. 缩放（Scaling）

图像的缩放可以分为两种情况，即等比例缩放和不等比例缩放。等比例缩放是指图像在 x 轴方向和 y 轴方向缩放的比例相同；不等比例缩放指在 x 轴和 y 轴方向进行不同比例的缩放，即图像的长宽比改变。设图像 x 轴方向的缩放比例为 α ， y 轴方向的缩放比例为 β ，则缩放前的像素坐标 (x,y) 与缩放后的坐标 (x',y') 之间的关系为

$$\begin{cases} x' = x \cdot \alpha \\ y' = y \cdot \beta \end{cases} \tag{9-2}$$

图 9-2 为对 Lena 标准图像进行缩放攻击的例子^[3]。可以看到，在图像受到缩放攻击的时候，图像的内容并没有发生变化。这里需要注意的一点是在图像被缩放的过程中存在插值操作，且当图像被缩小时由于图像分辨率的降低，会丢失原始图像中的一部分信息。换句话说，对图像进行的缩放攻击是一个不可逆的操作，尤其是图像被缩小的时候，部分原始信息的丢失会对水印系统造成很大的影响。此外，在不等比缩放中，图像的内容会发生变形。



(a) 原始图像 (b) 等比缩放 0.75 倍 (c) 等比缩放 1.2 倍 (d) 不等比缩放

图 9-2 图像缩放

3. 平移（Translation）

平移操作是将数字图像内的像素沿直线方向从一个位置移动到另一个位置。若设 x 轴和 y 轴方向的平移量分别为 x_0 和 y_0 ，则平移前像素坐标 (x,y) 和平移后的坐标 (x',y') 之间的关系为

$$\begin{cases} x' = x + x_0 \\ y' = y + y_0 \end{cases} \tag{9-3}$$

图 9-3 所示为图像平移的例子^[3]。由于平移只是像素空间位置的改变，所以并不会修改图像的像素值。因此，与图像的旋转和缩放操作不同，在图像平移的过程中不会引入插值误差。但需要注意的是，由于在图像平移的过程中会有一部分图像内容移出，故为了保证图像大小不变，往往需要将移出的部分补零。



(a) 原始图像

(b) 向右下平移 40 像素后的图像

图 9-3 图像平移

除了单独的旋转、缩放、平移攻击外，在数字水印系统中还可能在上述三种攻击的组合攻击，这些攻击可用如下的仿射变换模型来表示：

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = s \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \quad (9-4)$$

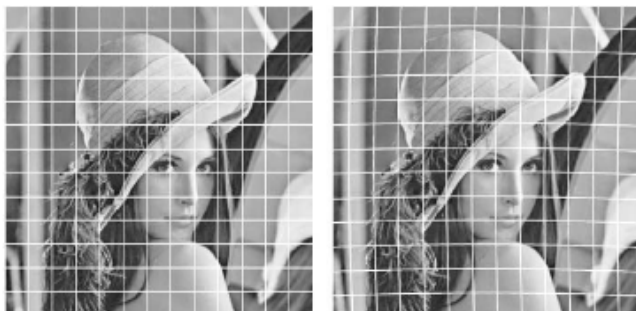
另外，还有翻转（Flipping）、拉伸扭曲（Shearing）、投影变换（Projective Transformation）等全局几何变换，恶意的攻击者也可能设计出各种各样的全局几何攻击企图去除水印信息。

9.1.2 局部几何攻击

与图像的全局几何攻击不同，局部几何攻击是指对图像的不同区域进行不同的攻击。常见的局部几何攻击有 Stirmark 随机扭曲、剪切、行/列删除等。

1. 随机扭曲攻击（RBA）

RBA 最初是由 Stirmark 软件平台引入的。其基本思想是将某个像素点的位置进行一定的位移，位移量是一个随机的量，一般为像素点坐标的函数。RBA 所引起的图像局部变形可用图 9-4 中的网格来表示^[3]。借助于网格，我们可以看出图像的各个部分受攻击的情况。但是在没有标记出网格的情况下，人眼是很难察觉到这种失真的。



(a) 原始图像

(b) RBA 图像

图 9-4 对图像实施RBA的效果

通常，RBA 包括随机移位（Random Displacement）攻击、整体扭曲（Global Bending）攻击和高频扭曲（High Frequency Bending）攻击。随机移位攻击又称随机抖动攻击，是将像素点的坐标改变一个随机的量。整体扭曲攻击和高频扭曲攻击是将原始像素点的坐标位移一个由坐标正弦函数确定的量。对整体扭曲攻击而言，图像中心位置的位移量比边界区域要大。图 9-5 给出了在 Lena 图像上进行这几种攻击的效果^[3]。



图 9-5 随机扭曲攻击

2. 剪切攻击

剪切攻击是指攻击者从含水印图像中裁减出一部分感兴趣的区域，同时丢弃不感兴趣的区域的过程。剪切攻击对图像造成的破坏是不可逆的，因为在水印检测时是不可能找回攻击者丢弃的那部分图像内容的。行/列删除攻击的作用与剪切类似。图 9-6 为剪切攻击和行/列删除的例子^[3]。一个好的水印算法应该能够在图像受到剪切攻击却保留主要的图像信息前提下进行有效的水印提取。



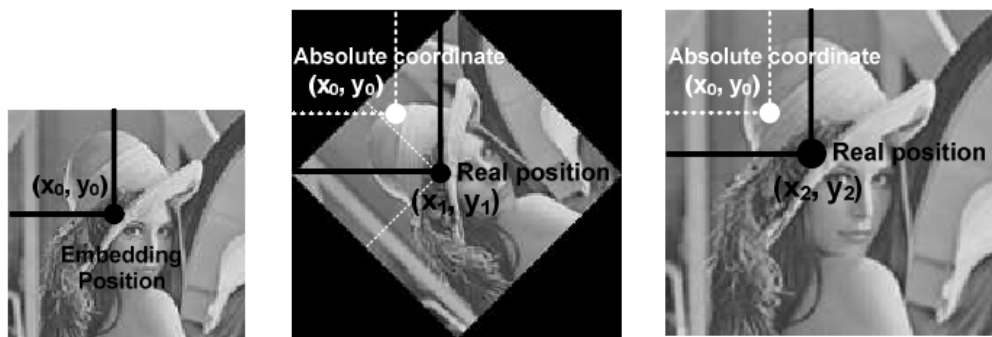
图 9-6 图像剪切

9.2 几何攻击对数字水印系统的影响

常规的信号处理攻击主要是削弱水印信号的能量，而几何攻击则会破坏水印嵌入与提取的同步性。例如当嵌入的水印信号为二值图像时，在受到信号处理攻击后，提取的水印信号可能会变得模糊而不容易辨别；而发生几何攻击之后则可能完全提取不出来水印。因此，几何攻击对水印系统的破坏是致命的。这个问题类似于通信系统中的同步问题，在通信系统中，

同步一旦丢失,则很难成功地进行通信。数字水印系统中的同步与通信系统类似,其直接影响水印信号的提取。因此为了抵抗几何攻击,水印算法必须首先保证嵌入和提取的同步性。

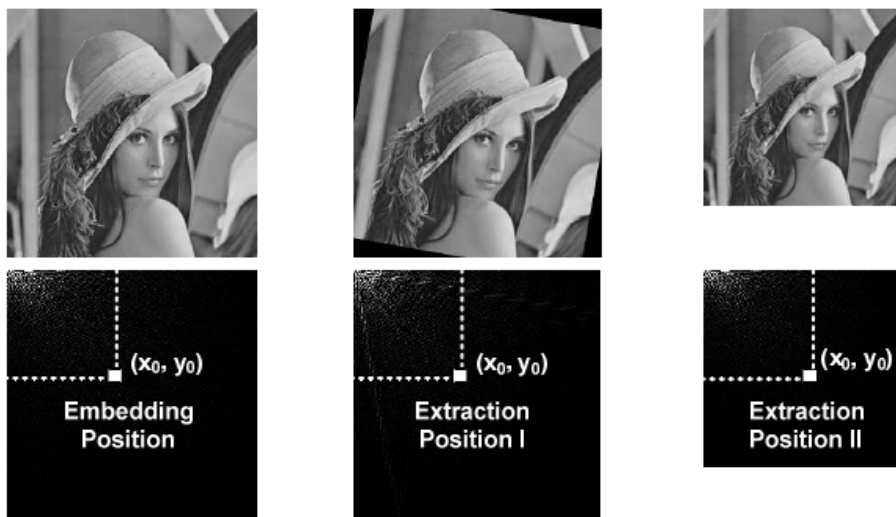
几何攻击对数字水印系统的影响可以用图 9-7 和图 9-8 说明^[3]。



(a) 原始图像及水印嵌入位置 (b) 旋转 45 度图像及水印提取位置 (c) 缩放 1.4 倍图像及水印提取位置

图 9-7 空域水印嵌入与几何攻击下的水印提取

图 9-7 可以说明几何攻击对空域水印算法的影响^[3]。假设在图 9-7 (a) 中,我们将一比特的水印信息嵌入像素 (x_0, y_0) 的邻域内(图中黑色圆区域),图 9-7 (b) 和图 9-7 (c) 分别是对含水印图像进行 45 度旋转和 1.4 倍放大后的图像。对于常规的水印算法,水印提取位置应与水印嵌入位置严格一致,故应在图 9-7 (b) 和图 9-7 (c) 中的绝对坐标 (x_0, y_0) 所确定的邻域内(图中白色圆区域)。但是由于受到了几何攻击,含有所嵌入水印比特的区域并不在上述绝对坐标 (x_0, y_0) 的邻域内,而分别变到了图 9-7 (b) 中坐标 (x_1, y_1) 的邻域和图 9-7 (c) 中坐标 (x_2, y_2) 的邻域(图中的黑色圆形区域)。因此,如果按照常规的水印提取方法,在受到几何攻击之后无法准确提取水印。



(a) 原始图像及水印嵌入位置 (b) 旋转 10 度图像及水印提取位置 (c) 缩放 0.5 倍图像及水印提取位置

图 9-8 DCT 域水印嵌入与几何攻击下的水印提取

几何攻击对变换域水印算法的影响可用图 9-8 说明。假设我们在原始图像 DCT 系数

(x_0, y_0) 处嵌入一比特水印信息, 则对于一般的水印算法, 水印提取时应该也在测试图像 DCT 变换的坐标 (x_0, y_0) 系数处, 如图 9-8 (b) 和图 9-8 (c) 所示。但是由于测试图像相对于原始含水印图像已经发生了几何变换, 所以测试图像的 DCT 变换在坐标 (x_0, y_0) 处的系数值已经发生了很大的变化。因此, 在水印嵌入位置 (x_0, y_0) 处就不可能成功提取水印。

基于以上分析可知, 要使水印算法能够抵抗几何攻击, 必须首先保证水印嵌入和水印提取在相同的位置进行。这一过程在抗几何攻击水印技术中被称为水印同步, 其基本思想与通信系统的信号同步类似。

9.3 抗几何攻击的数字水印技术

根据 9.2 节的分析我们知道, 要使水印能够抵抗几何攻击, 必须首先保证水印嵌入和水印提取在相同的位置进行, 即水印同步。目前, 水印同步的思路主要有以下三种^[2~5]。

1. 基于几何校正的方法

在水印检测之前首先消除几何变换的影响, 即对受到几何攻击的含水印图像首先进行逆变换, 再用常规的方法提取水印。这类方法的关键在于在水印提取之前准确计算出图像受到的几何攻击参数, 如旋转的角度、缩放的倍数等。已有算法中的穷尽搜索法、基于图像配准的方法和基于模板嵌入的方法都属于这一类。

2. 基于不变域的水印嵌入

基本思想是在水印嵌入和提取之前首先找到一种对几何攻击具有不变性的域。基于傅里叶-梅林变换的方法、基于不变矩的方法和基于图像归一化的方法都属于这一类型。

3. 基于图像特征的水印同步

将水印信息与从图像当中提取的具有几何不变性的特征相联系来实现水印嵌入和提取。常用的特征包括图像的边缘、特征点、图像的统计特征和具有几何不变性的区域等。

9.3.1 基于几何校正的方法

基于几何校正的抗几何攻击数字水印算法主要包括穷尽搜索法、基于图像配准的方法和基于模板嵌入的方法。

1. 穷尽搜索法

穷尽搜索法的思想很简单, 就是对含水印图像所有可能经受的几何变换进行逆变换后进行水印提取。如果水印检测的时候可以借助于原始图像, 则可以对比分析出攻击前后含水印图像与原始图像的几何关系, 估计出含水印图像的几何变换参数, 在水印检测之前以一定的精度对图像进行相反的几何变换, 可以大大提高水印检测的准确率。

原始图像的信息很多, 根据使用信息的数量不同, 也带来了不同的利用原始图像抗几何

攻击的方法。文献[6]利用整个原始图像为参考,对攻击后图像进行校正。文献[7]比较了整个图像攻击前后的异同,引入运动估计的模型,对失真情况进行校正。而文献[8]只利用了一部分图像和密钥的配合,来完成图像的恢复工作。在更多情况下,只需要图像的一部分线特征或点特征,就可以进行图像的恢复,文献[9, 10]提出了一种基于广义 Radon 变换和特征点的抗几何失真水印算法,引用了两种广义 Radon 变换来抵抗缩放攻击和旋转攻击,通过图像特征点来抵抗平移攻击。

由于几何变形是无法预知的,因此几何变形的类型和参数的估计只能采取搜索估计-判断的策略,这是一个非常随机的过程,而且计算量较大。对每一种可能的几何攻击,都需要按照一定的步长做逆变换,然后判断逆变换后图像与原始图像的相似情况,以此决定是否继续进行几何变形和相应的变换参数。当有多个几何攻击同时存在的时候,工作将很难进行。并且,利用原始图像的方法是一种非盲的检测算法,需要提供未受攻击的图像,这大大限制了其使用范围。

2. 基于图像配准的方法

在非盲检测的水印算法中,可以借助于原始图像或者未受攻击的含水印图像,采用图像配准(Image Tration)技术^[11]估计出水印图像的几何变形及参数,以获得水印同步。其主要步骤如图 9-9 所示^[3]。

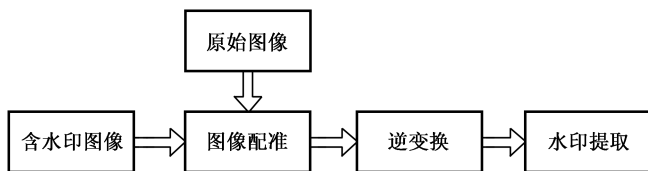


图 9-9 基于图像配准的水印检测框图

对于含水印图像与原始图像,首先利用图像配准技术求得几何变换参数,然后对受攻击的含水印图像进行校正,最后提取水印。图像配准本质上就是分析待配准图像上的几何畸变然后采用一种几何变换将图像变换到统一的坐标系统中的技术,在医学图像处理和计算机视觉中有着广泛的应用。图像配准中得到的几何变换参数可以用于对受攻击的含水印图像进行校正。图像配准技术可以分为基于区域的方法^[12, 13]和基于特征的方法^[14, 15],如图 9-10 所示^[3]。

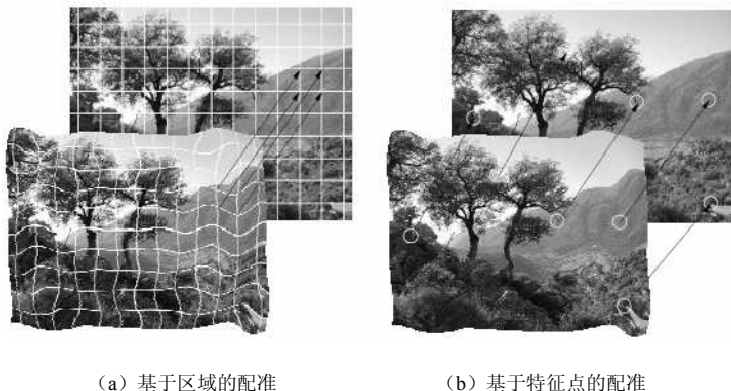


图 9-10 图像配准

基于区域的配准算法，一般在进行处理之前，首先定义区域的大小，并将参考图像划分为若干个分离的临时区域，然后通过改变目标图像上移动区域的位置和大小，来计算它与参考图像上所选区域的相关值，再利用循环比较的计算方法来搜寻一定范围内有最大相关值的位置，其所在位置的坐标和两区域比例因子的大小作为配准公式参数。这类算法直接利用图像的像素信息，因此抗噪声能力差，对旋转攻击处理比较困难，并且计算量大，配准效率低。在基于图像几何特征的配准算法中，算法的稳定性主要依赖于特征空间的选取（点特征、线特征、区域特征等）、相似性度量、搜索空间、搜索策略等。从实际情况来看，图像几何特征的数量与位置的精确定位通常受限制，因为图像中的几何基本体经常由于过于简单而导致图像几何特征稀少和不精确。另外，在原始图像中嵌入水印后，图像的特征点往往会偏移 1 至 2 个像素，这容易导致配准精度的下降^[16]。

另外，在水印提取的过程中需要原始的未加水印的图像参与，或者需要在海量数据库中寻找未受攻击的含水印图像，因而其应用受到限制。

3. 基于模板嵌入的方法

基于模板嵌入的方法与基于图像配准的方法不同之处在于不是对图像进行配准，而是对模板进行配准，通过模板匹配得到几何变换参数，最后进行相应的逆变换后提取水印，其系统框图如图 9-11 所示^[3, 4]。

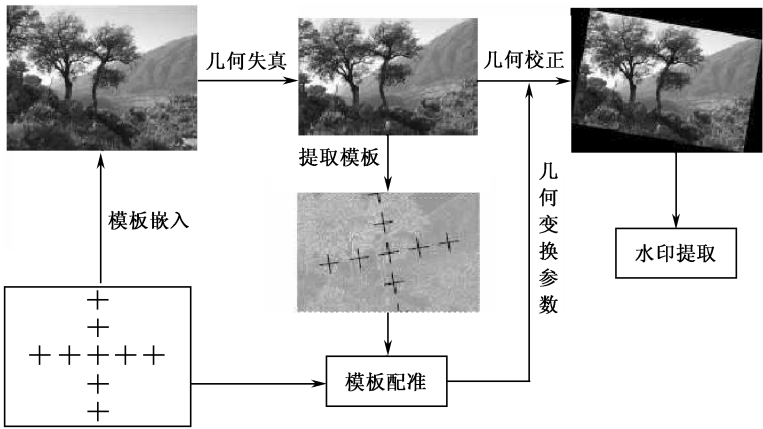


图 9-11 基于模板嵌入的水印算法框图

在这种方法中，对原始图像除了进行常规的水印嵌入步骤外，还须将预先设计好的模板嵌入原始图像中。在水印检测端，首先将嵌入的模板提取出来，并将其与原始模板进行配准，获得模板发生的旋转、缩放等几何变换参数，也就是含水印图像所受到的几何变换参数。然后根据几何变换参数对图像进行逆变换得到几何校正后的图像，最后进行水印提取。

基于模板嵌入的算法最重要的步骤就是如何将模板有效地嵌入图像中。模板信号通常为具有规则形状的标记，也称导航信号（Pilot Signal）。这样，当从受攻击的图像中提取出模板信号的时候就可以根据其形状判断出图像所经受的几何变换参数。模板信号可以在空域嵌入，也可以在频域嵌入。

但是，利用模板做同步工具，对几何攻击进行校正的方法存在几个问题。首先，嵌入的模板和水印信息虽然功能不同，但实质上都是嵌入的信息，都需要在攻击后的待测图像中进

行提取。这样实际上是嵌入了两份不同的水印信息，两个不同的嵌入信息之间会产生干扰，可能会导致模板和水印这两个信号中较弱的信号无法顺利地检测出来。其次，水印和模板同时嵌入必然增加了信息的嵌入容量，也对水印容量和嵌入作品的保真度提出了更高要求。另外，此类算法容易受到模板移除（Template Removal）攻击，因为在算法已知的情况下任何人都可以对模板进行操作，将其去除。

水印去同步攻击对任何水印方案来说是最有效的攻击方式之一。目前一个最多的争论点就是穷尽搜索和模板匹配是否是解决水印去同步攻击的有效方法。实际上，通过穷尽搜索的水印同步方法不但计算复杂，而且它会显著地增加错检率。对于模板匹配方法，必须考虑同步错误。

在文献[17]中，Barni 对该问题进行了研究并对两种方法进行了比较，结果表明，只要搜索空间不是呈指数方式增长，两种方法都是近似良好的。而且，从检测的可靠性来讲，穷尽搜索法要优于基于模板匹配的再同步法。

9.3.2 基于几何不变域的方法

基于不变域的水印嵌入是抗几何攻击水印算法中较早出现的一种。这类算法的出发点是从原始图像中寻找具有几何不变性的量来隐藏水印。由于载体本身就具有几何不变性，那么嵌入的水印在几何攻击后也可以保持不变。典型的几何不变域包括对数极坐标与傅里叶-梅林（Fourier-Mellin）变换域、图像不变矩和图像归一化域等。

1. 基于傅里叶-梅林变换的方法

O'Ruanaidh 等人于 1997 年提出了一种基于傅里叶-梅林变换的抗几何攻击图像水印算法^[18, 19]，该算法的框图如图 9-12 所示。对原始图像，首先进行离散傅里叶变换（DFT），并将幅频部分进行对数极坐标映射（LPM），然后再进行一次 DFT，并在其幅频中嵌入水印。水印嵌入后，先结合相位信息对修改后的系数进行一次逆傅里叶变换（IDFT），接着进行逆对数极坐标映射（ILPM），最后结合原始的相位信息进行一次 IDFT 得到加水印后的图像。

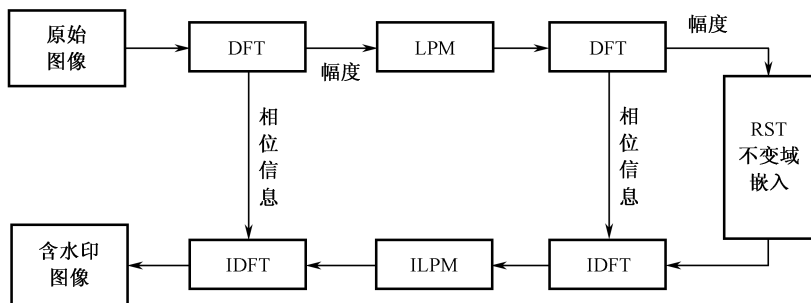


图 9-12 O'Ruanaidh 基于傅里叶-梅林变换的水印算法框图

在该算法中，离散傅里叶变换的平移不变性保证了水印的平移不变性，而旋转和缩放不变性则由对数极坐标变换后的傅里叶幅频特性保证。这一算法理论很完善，然而在实际实现中最大的问题就是会对加水印图像的质量造成较大的影响，这是因为在对数极坐标映射和逆对数极坐标映射的过程中存在较大的插值误差，如图 9-13 所示^[3]。

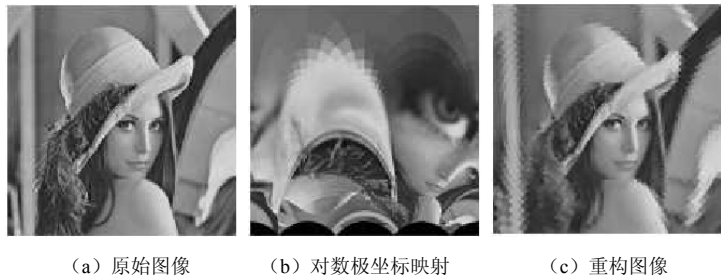


图 9-13 图像的对数极坐标映射与图像重构

针对傅里叶-梅林变换的 LPM 与 ILPM 过程中的插值误差，一些学者提出了改进的方法。这些改进算法的主要思路为避免 ILPM 这一过程。例如 Lin 等首先对原始图像进行 DFT，然后对幅频进行 LPM，然后沿着对数半径轴由幅频构造出一维信号，最后将水印嵌入其中^[20]。图像的旋转对应一维信号的周期平移，图像的缩放对应一维信号的缩放，而图像平移对一维信号没有影响。Zheng 等首先对图像进行 DFT，然后对其幅频信息进行 LPM，嵌入水印，并利用原始图像 LPM 与含水印图像 LPM 之间的相位相关性（Phase Correlation）来计算水印在 LPM 域中的位移^[21]。

2. 基于图像不变矩的方法

基于图像不变矩的方法主要是利用各种不变矩对图像旋转、缩放等几何变换的不变性进行水印嵌入。常用的图像矩包括 Zernike 矩、Krawtchouk 矩、Tchebichef 矩等。其中，基于 Zernike 矩的算法研究相对较多。Kim 等通过修改低于 5 阶的 Zernike 矩嵌入一比特的水印信息，并借助于图像归一化实现缩放和平移的不变性^[22]。水印的旋转不变性由 Zernike 矩的旋转不变性保证。Xin 等通过对特定阶数的 Zernike 矩进行抖动调制（Dither Modulation）嵌入水印序列，并借助于最小距离检测器提取水印^[23~26]。Xin 等的算法框图如图 9-14 所示。

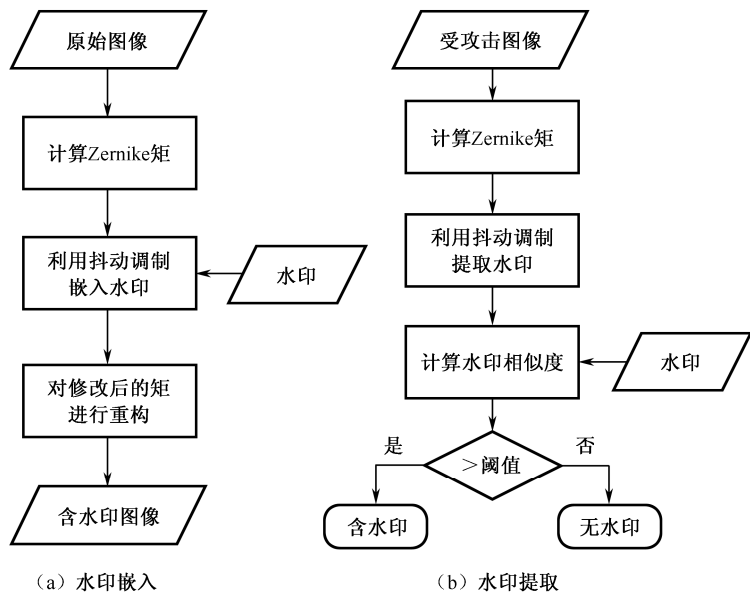


图 9-14 基于Zernike矩的水印算法框图

基于 Zernike 矩的水印算法的缺点是容易在图像中矩的计算区域边界造成较明显的失真。由于 Zernike 矩的计算区域一般取图像中的内接圆部分，因此利用 Zernike 矩加水印之后容易产生“圆环效应”，如图 9-15 所示^[3]。



(a) 原始图像

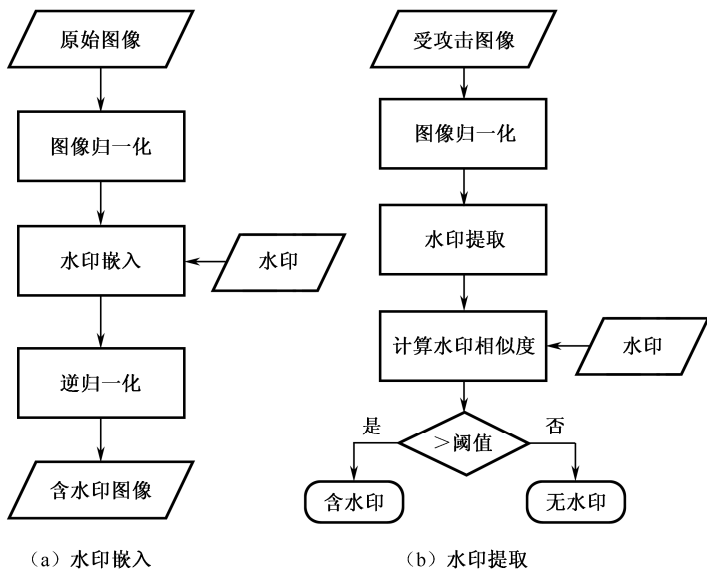
(b) 含水印图像

(c) 放大的差值图像

图 9-15 基于 Zernike 矩的水印嵌入效果

3. 基于图像归一化的方法

图像归一化是将图像变换到统一的大小和方位的技术，在模式识别中有着广泛的应用。图像的归一化参数可由图像的几何矩计算^[27]。将图像归一化技术应用于数字水印中，其基本思想是在水印嵌入和水印提取之前先对图像进行归一化，消除几何变换的影响。此类方法的水印嵌入和水印提取的一般框图如图 9-16 所示。



(a) 水印嵌入

(b) 水印提取

图 9-16 基于图像归一化的水印算法框图

基于图像归一化的抗几何攻击数字水印算法存在的主要问题有以下两点。

① 水印嵌入过程需要先对原始图像进行归一化得到几何不变域，接着嵌入水印，最后还需要对图像进行逆归一化得到含水印图像。在图像的正向和反向归一化过程中存在插值运算，因此在水印嵌入的过程中就会引入一定的插值误差，其作用类似于常规的信号处理攻击。

② 图像归一化参数的计算是基于图像几何矩的，而几何矩的计算是基于图像的整体内容

的。所以，当图像受到剪切、平移等存在信息丢失的攻击时，几何矩的计算会产生很大的误差，进而直接影响图像归一化的精度。因此，基于图像归一化的数字水印算法很难抵抗图像平移、剪切、行/列删除等攻击。

9.3.3 基于图像特征的方法

前面提到的抗几何攻击水印算法，不论是空间域方法还是频域方法，都只考虑怎么在像素、频率或变换系数中嵌入水印信息，而忽略了图像本身的特征。图像的特征往往代表图像最本质的特性，而且这些特性不管是在受到常规的信号处理攻击还是几何攻击时都保持相对稳定的状态。因此，如果能将水印信息与这些图像特征相联系，就能保证水印对于这些攻击的不变性。第二代数字水印——基于图像特征的水印算法正是利用这一思想来实现水印嵌入和水印提取的同步性的。

第二代数字水印是建立第一代的基础上的，也就是说其用到的一些基本原理、理论、方法与第一代相同，而且仍须具备第一代水印的鲁棒性。除此之外，两个重要的不同点是：

- ① 第二代水印利用图像自身的稳定特征来确定水印的嵌入位置；
- ② 第二代水印增强了数字水印对几何攻击的抵抗能力。

水印算法中利用的稳定特征可以是抽象的，也可以是语义上有意义的，如图像的边缘、边角点、纹理区域，或者是图像中具有一定特征的部分。在图像的众多特征当中，特征点由于其提取简单、稳定性好等优点，在抗几何攻击数字水印算法中得到了广泛的应用。

常用的特征点提取方法包括基于墨西哥帽小波的特征点^[1]、SUSAN 特征点^[28]、Harris 特征点^[29]、Harris-Laplace 特征点^[30, 31]、尺度不变特征变换 (SIFT)^[32]等。SUSAN 特征点和 Harris 特征点的检测过程是在单一尺度下进行的，即在图像本身所在的尺度进行检测。因此，我们称这些特征点为传统的特征点。与此不同的是，Harris-Laplace 特征点和 SIFT 特征点检测过程基于图像的尺度空间理论^[33]，这些特征点都与特定的尺度相对应，即表示该特征点所在的尺度，这一类特征点称为尺度空间特征点或多尺度特征点。传统的特征点由于检测尺度的单一性，往往对图像的尺度变换不具有不变性。换句话说，在原始图像中检测到的特征点在缩放后的图像中往往不能够检测出来。与此不同的是，多尺度特征点在图像的尺度空间中检测特征点，即每一个特征点都与一个特定的尺度相对应。所以，多尺度特征点除了对图像的旋转具有不变性外，对图像的尺度变换也具有不变性。因此，借助于多尺度特征点可以在图像中自适应地确定局部区域，以达到水印同步的目的。

但是并非所有特征都适于嵌入水印，图像的均匀区域就不适于嵌入水印，因为图像的伸缩处理会引起这部分的变化。Kutter 给出了适于作为水印同步的特征点应该具备的几个条件^[1]：

- ① 具有对噪声的不变性，如 JPEG 压缩、加性或乘性噪声；
- ② 具有对几何变换的不变性，即对图像的旋转、平移、下采样等操作具有不变性；
- ③ 应该具有局部化特性，即对图像的剪切操作不会影响剩余的特征点。

另外，Kutter 还指出利用图像的特征点进行水印算法设计的两类方法：

- ① 以提取的特征点为参考进行水印嵌入；
- ② 在所提取的特征点中直接进行水印嵌入。

目前，已有的此类算法都采用第一种方案，并且已有算法中所采用的局部区域主要是三

角形和圆形,如图 9-17 所示^[3]。



图 9-17 基于特征点的局部区域

图 9-17 (a) 所示为对提取的特征点进行 Delaunay 三角剖分生成的相互连接的三角形区域,图 9-17 (b) 为采用传统的特征点确定的大小相同且互不重叠的圆形区域,图 9-17 (c) 为采用多尺度特征点在图像中自适应地确定的圆形区域。确定三角形区域时若采用传统的特征点,则三角形区域理论上仅具有旋转不变性而不具有缩放不变性;若采用多尺度特征点,则三角形区域具有旋转和缩放的双重不变性。此外,一个三角形区域的确定需要三个特征点的信息,只要其中的一个特征点检测错误就会导致三角形区域变形。因此,基于三角形区域的水印同步对特征点的鲁棒性要求较高。基于圆形区域的水印同步在确定圆形区域时只需要一个特征点,并且圆形区域在图像发生旋转时内部的内容保持不变。此外,相对于传统特征点确定的圆形区域,基于尺度空间特征点的圆形区域由于具有旋转和缩放的双重不变性,可以更加有效地实现水印同步。需要注意的是,水印同步时需要保证圆形区域之间互不重叠,否则嵌入的水印信息会相互影响。

9.4 基于 Harris 特征点的抗几何攻击的数字图像水印算法

在基于图像特征点的抗几何攻击水印算法中,Bas 等提出的基于 Harris 特征点和 Delaunay 三角剖分的算法是其中的经典方法之一^[34]。

9.4.1 Harris 特征点检测

Harris 特征点检测器^[29]是 Harris 和 Stephens 在 Moravec 算法^[35]基础上发展起来的,Moravec 算法的思想是:在图像中设计一个局部滑动窗,当该窗沿各个方向做微小移动时,考察窗口的平均能量变化,当该能量变化值超过设定的阈值时,就将窗口的中心像素点提取为角点。

Harris 特征点检测器用自相关函数来确定图像信号发生二维变化的位置。对一幅数字图像 $f(x,y)$,Harris 特征点的主要的检测步骤如下。

① 计算图像 $f(x,y)$ 在 x 轴方向和 y 轴方向的梯度:

$$\begin{cases} X = f(x,y) * [-1,0,1] = \partial f(x,y) / \partial x \\ Y = f(x,y) * [-1,0,1]^T = \partial f(x,y) / \partial y \end{cases} \quad (9-5)$$

其中，*表示卷积。

② 构造自相关矩阵。

令

$$\begin{cases} A = X^2 * w \\ B = Y^2 * w \\ C = (XY) * w \end{cases} \quad (9-6)$$

其中， $w = \exp(-(X^2 + Y^2)/2\sigma^2)$ 为高斯平滑窗函数。由式 (9-6) 可以得到图像 $f(x,y)$ 的自相关矩阵：

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial f(x,y)}{\partial x}\right)^2 & \left(\frac{\partial f(x,y)}{\partial x}\right)\left(\frac{\partial f(x,y)}{\partial y}\right) \\ \left(\frac{\partial f(x,y)}{\partial x}\right)\left(\frac{\partial f(x,y)}{\partial y}\right) & \left(\frac{\partial f(x,y)}{\partial y}\right)^2 \end{bmatrix} \quad (9-7)$$

③ 提取特征点。

对自相关矩阵 M 首先计算其行列式的值和迹：

$$\begin{cases} \text{Det}(M) = AB - C^2 \\ \text{Trace}(M) = A + B \end{cases} \quad (9-8)$$

Harris 检测器的响应由下式计算：

$$R_H = \text{Det}(M) - k \cdot \text{Trace}^2(M) \quad (9-9)$$

其中， k 为常数，通常取 0.04~0.06 之间。最后，将检测器响应 R_H 与一阈值 T 比较即可检测出特征点的位置。并且，阈值 T 设置得越大，则检测出的特征点数量越少。此外， R_H 的值与特征点的鲁棒性还有着直接的关系，响应值越大表明特征点的鲁棒性越好。

图 9-18 所示为在 Lena 图像上进行 Harris 特征点检测的结果^[3]。从图中可以看出，Harris 特征点主要集中在图像的纹理复杂区域和有明显灰度变化的地方。



图 9-18 Lena 图像 Harris 特征点检测结果

9.4.2 Delaunay 三角剖分

离散点的三角剖分是科学计算与分析中的一种重要方法,在计算几何、数据插值曲面构成和图形学方面得到了广泛的应用。对离散点的三角剖分方法的研究不论是在二维平面还是在三维空间区域上都取得了很多成果,尤其是对二维平面离散点的研究,理论和算法已经比较成熟。而 Delaunay 三角剖分,因为本身具有一些很好的性质,至今仍是许多学者研究的内容。

由 Delaunay 三角剖分方法生成的三角网——Delaunay 三角网,是一种特殊的三角网,下面给出两个相关的定义。

☒ 定义 9-1 Delaunay 边

假设一条边 l (l 的两个端点为 a 和 b , 其中 $a, b \in N$), 如果存在一个圆经过 a 和 b 两点, 而在该圆内, 不包含集合 N 中的任何其他点, 则称 l 边为 Delaunay 边, 该特性称为空圆特性。

☒ 定义 9-2 Delaunay 三角剖分

如果点集合 N 的三角剖分只包含 Delaunay 边, 则该三角剖分称为 Delaunay 三角剖分, 形成的三角网称为 Delaunay 三角网。

Delaunay 三角网具有许多好的特性:

① Delaunay 三角网在特征点均匀分布的情况下以最临近的三点形成三角形, 各线段不相交, 在三角网中的任意一个三角形内也不会包含其他的三角形。三角网中没有相交、包含的三角形, 可以保证三角网覆盖特征点涵盖的所有区域, 并且剖分的图像区域没有重复部分。

② 不论从图像的任何地方开始构造 Delaunay 三角网, 只要特征点没有改变, 始终得到的结果都唯一。唯一性大大降低了构造三角网的复杂性, 这种特性更适合在遭受几何攻击后, 失去同步的情况。

③ Delaunay 三角网形成的三角形的最小角最大, 这样就可以避免产生狭小和过小锐角的三角形, 并且这样剖分产生的三角形最接近规则三角形。

④ 如果新增特征点或者某个特征点消失, 则只影响临近的三角形, 与其他三角形无关。

从 Delaunay 三角形的特性可以看出, Delaunay 三角形具有很好的局部稳定性, 也保证了剖分形成的三角形具有较高的水印容量。现在有许多三角网生成算法^[38~40]。

9.4.3 基于 Harris 特征点和 Delaunay 三角剖分的水印算法

本节介绍 Bas 等提出的基于 Harris 特征点和 Delaunay 三角剖分的水印算法。该算法流程如图 9-19 所示。首先在原始图像上进行 Harris 特征点检测, 接着利用 Delaunay 三角剖分将图像分割成相互连接的三角形区域, 然后将水印嵌入每一个区域当中。具体嵌入时, 首先将原始的水印信号变换到与各个三角形相同的大小和方向, 然后在一定的视觉掩蔽特性的作用下在空间域叠加到原始的区域中实现水印嵌入。在水印检测时, 首先利用与水印嵌入过程相同的方法生成三角形区域, 然后在每一个三角形当中提取水印信号, 并计算与原始水印的相关

系数来判断水印存在与否。

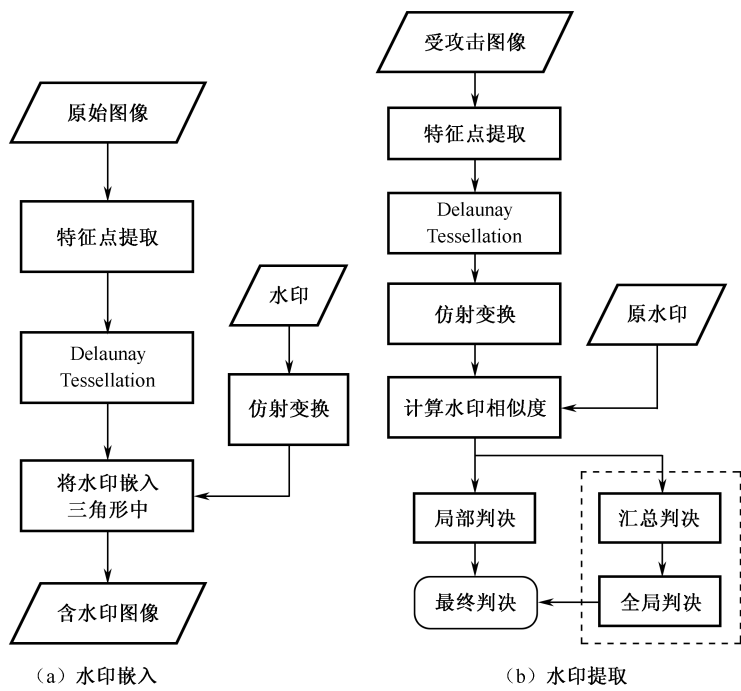


图 9-19 Bas等提出的水印算法框图

9.5 基于 SIFT 特征点的抗几何攻击的数字图像水印算法

9.5.1 尺度空间理论

近些年，学者们对基于图像特征点的抗几何攻击数字水印技术进行了大量的研究。常规的特征点检测算法，如 Harris 特征点、SUSAN 特征点等，都是在单一尺度上进行检测，因此这些特征点本身并不具有缩放的不变性。换句话说，在原始图像上能检测到的特征点当图像发生缩放以后并不一定能检测出来。如果直接以这些特征点作为参考进行水印嵌入，则所设计的水印算法很难抵抗缩放攻击。近年来，模式识别领域中尺度空间特征点的出现给这一问题带来了解决办法。尺度空间的特征点，如 Harris-Laplace 特征点^[30, 31]和尺度不变特征变换 (SIFT)^[32]，一般具有对图像旋转、缩放、平移的多重不变性，因此为设计具有 RST 不变性的数字水印算法提供了理论依据。

尺度空间是一种特殊的多尺度表达的方式，由一个连续的尺度参数确定且在所有尺度上均保持了相同的空域采样^[33]。一个信号的尺度空间表达方式是将该信号与宽度逐层增大的核函数相卷积的结果。并且，已有的研究表明在一定的假设条件下唯一可能的尺度空间核函数为高斯函数。

一般来说，一个连续信号的尺度空间表达方式可以这样来构造^[33]：令 $f: R^N \rightarrow R$ 表示一

个信号, 则其尺度空间表示 $L: R^N \times R_+ \rightarrow R$ 定义为 $L(\cdot; 0) = f$ 并且

$$L(\cdot; t) = G(\cdot; t) * f \quad (9-10)$$

其中, $*$ 表示卷积, $t \in R_+$ 为尺度参数, $G: R^N \times R_+ \setminus \{0\} \rightarrow R$ 为高斯核函数, 可以表示为

$$G(\mu; t) = \frac{1}{(2\pi t)^{N/2}} e^{-\mu^T \mu / (2t)} = \frac{1}{(2\pi t)^{N/2}} e^{-\sum_{i=1}^N \mu_i^2 / (2t)} \quad (9-11)$$

其中 $\mu \in R^N$ 并且 $\mu_i \in R$ 。尺度参数的平方根, 即 $\sigma = \sqrt{t}$, 为高斯核 G 的标准差, 也是在尺度 t 下平滑后图像的空间尺度的自然量度。

对于数字图像, 尺度空间定义为将可变尺度的高斯函数 $G(x, y, \sigma)$ 与输入图像 $I(x, y)$ 的卷积, 若记尺度空间为 $L(x, y, \sigma)$, 则有

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (9-12)$$

其中

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (9-13)$$

图 9-20 所示为从 Lena 标准图像的尺度空间表达中取出的几幅图像^[3]。由上而下, 由左及右, 对应的高斯函数的尺度逐步增大。可以看出随着高斯核函数方差的增大, 图像越来越模糊。



图 9-20 标准图像Lena尺度空间中的图像

9.5.2 SIFT 特征点

2004 年, 哥伦比亚大学的 David Lowe 提出了一种新的特征点提取算法——SIFT 算法^[32], 较好地解决了场景部分遮挡、旋转缩放、视角变化引起的图像变形等问题, 并且成功应用在了目标识别、图像复原、图像拼接等领域。

SIFT 变换提取特征点的基本思想是利用一系列的滤波来提取图像尺度空间中的稳定点。相对于 Harris-Laplace 特征点, SIFT 特征是图像的局部特征, 不仅对旋转、尺度缩放、亮度

变换保持不变性，而且对视角变化、仿射变换、噪声也保持一定程度的稳定性。SIFT 特征点的检测是通过一种分级的方式，通过在不同尺度下特征点的提取，比较选择出尺度变换后的稳定点，再去掉边缘和特征不明显的特征点，构成了 SIFT 特征点集。按照 David Lowe 的算法，这个过程主要通过四步完成。

1. 把平面图像立体化，建立图像的尺度空间，寻找备选点

为了在尺度空间中有效地检测出特征点的位置，Lowe 等提出在高斯差（Difference of Gaussian, DoG）函数中判断尺度空间的极值。DoG 函数 $D(x,y,\sigma)$ 可以通过对尺度空间中尺度间隔为 k 的两个图像求差获得。

$$D(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) = L(x,y,k\sigma) - L(x,y,\sigma) \tag{9-14}$$

k 表示尺度变化的系数。实验发现，即使尺度有比较大的改变，也不会影响极值点的判定和点位置的确定，文献[41]中给出了一个经验参数， $k = \sqrt{2}$ 。

如果将多个相邻尺度的图像相减，就可以得到一组立体的尺度空间图像集 $\{D(x,y,m\sigma)\}$ 。然后将图像缩小 2 倍，再重复上述过程，计算方法跟在原尺寸上的计算方法相同，只是尺寸缩小后计算量大幅减少。继续计算下一组的尺度空间图像，直至图像尺寸小于某一范围，例如 16×16 。

$$D(x,y,m\sigma) = (G(x,y,k^m\sigma) - G(x,y,k^{m-1}\sigma)) * I(x,y), \quad m = 1, \dots, s, k = 2^{1/s} \tag{9-15}$$

式中， $D(x,y,m\sigma)$ 表示相邻尺度的 DoG 图像，为了保证每组尺度空间图像上的极值点判别过程可以覆盖该倍程所有的点，每组必须生成 $s+3$ 幅图像。在实际计算中， $k = \sqrt{2}$ ，所以 $s=2$ ，每个倍程须生成 5 幅图像，整个过程如图 9-21 所示。

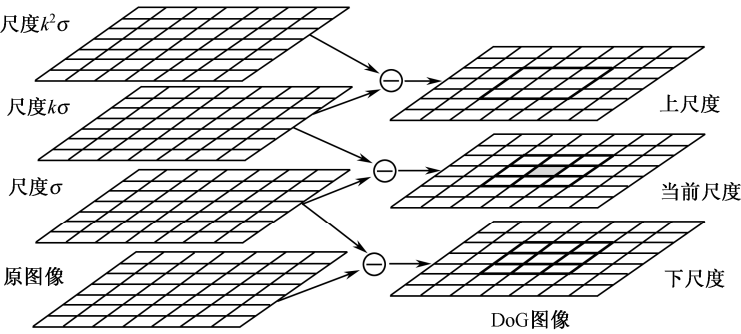


图 9-21 图像尺度空间及DoG函数

图 9-21 描述了 DoG 函数的生成过程。对原始图像，首先利用不同差分高斯核函数与其卷积产生不同尺度下的尺度空间图像，如图左边所示。然后将尺度空间中相邻的图像相减，就产生了 DoG 图像，如图右边所示。在检测尺度空间极值时，将 DoG 图像中的每一个点与同一尺度的周围领域 8 个像素和相邻尺度对应位置的周围领域 9×2 个像素共 26 个像素进行比较，以确保在尺度空间和二维图像空间上都检局部极值。

2. 筛选特征点

初始的特征点确定之后，利用一个二次函数来确定特征点的精确坐标和特征尺度。另外，对低对比度的点和处于边缘处的特征点，利用 Hessian 矩阵来判断其稳定性，Hessian 矩阵 H 定义为

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (9-16)$$

如果极值点满足

$$\frac{\text{Tr}(H)^2}{\text{Det}(H)} < \frac{(\alpha + \beta)^2}{\alpha\beta} \quad (9-17)$$

$$\text{Tr}(H) = D_{xx} + D_{yy} \quad (9-18)$$

$$\text{Det}(H) = D_{xx}D_{yy} - (D_{xy})^2 \quad (9-19)$$

则该极值点保留，否则该点将会被排除。其中， α 和 β 是 Hessian 矩阵最大的特征值和最小的特征值。

3. 为每个选定的特征点指定方向

SIFT 算法利用特征点邻域像素的梯度方向分布特性为每个特征点指定主方向，也就是特征点邻域内各点梯度方向的直方图中最大值所对应的方向。为了使检测的特征点具有旋转不变性，利用每个特征点邻域像素的梯度方向分布特性为其指定方向参数。梯度的模和角度可以表示为

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (9-20)$$

$$\theta(x, y) = \tan^{-1} \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \quad (9-21)$$

其中 $L(x, y) = G(x, y, \sigma) * I(x, y)$ 。

4. 生成描述符

SIFT 特征点检测的最后一步是对每一个检测出来的特征点生成一个描述符，该描述符可以用于特征点的匹配。

将关键点为中心的 16×16 矩形窗口，均匀地分成 16 个 4×4 的子块。在每个子块上计算 0、45、90、135、180、225、270、315 这 8 个方向的梯度累加值，绘制梯度方向直方图。16 个子块生成 16×8 共 128 个数据，这个 1×128 的向量就构成了特征点的描述符，并可以将描述符归一化，使其对亮度变化不敏感。

图 9-22 用一个 8×8 的窗口做了演示。

图 9-22 中黑点表示当前特征点所在的位置，以特征点为中心取 8×8 的窗口，每个小格代表特征点邻域所在的尺度空间的一个像素，箭头方向代表该像素的梯度方向，箭头长度代表梯度的模值，越靠近特征点的像素梯度方向信息贡献越大。然后在每 4×4 的小块上计算 8 个方向的梯度方向直方图，绘制每个梯度方向的累加值，即可形成一个种子点，如图 9-22 (b) 所示。图中的一个特征点由 4 个种子点组成，每个种子点有 8 个方向向量信息。这种邻域方向性信息联合的思想增强了算法抗噪声的能力，同时对于含有定位误差的特征匹配也提供了较好的容错性。

从特征点的生成方法可以看出，得到的 SIFT 特征点除了具有旋转和平移不变性，也具有很好的尺度不变性。

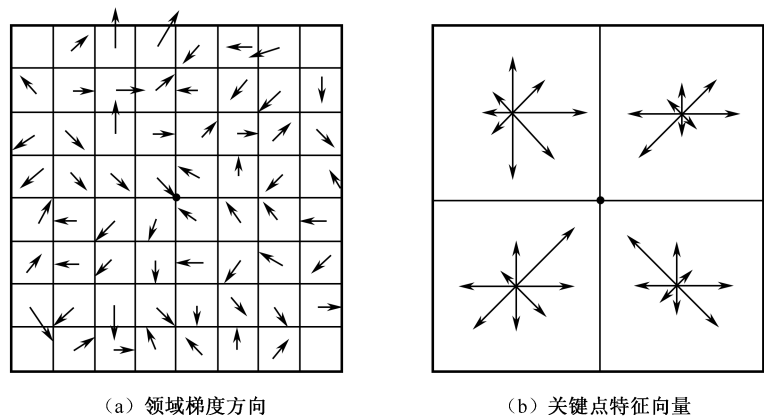


图 9-22 由特征点邻域梯度信息生成特征向量

通过以上四步，可以完成整个 SIFT 特征点的检测过程。图 9-23 显示从 Lena 图像中提取出的 SIFT 特征点^[3]。



图 9-23 SIFT算法提取的特征点

9.5.3 基于 SIFT 的水印同步

本节主要介绍 Lee 等提出的一种基于 SIFT 特征点的水印同步方法^[42]，Lee 等利用尺度不变特征变换（SIFT）提取图像的多尺度特征点，并将预先确定的水印信号经过适当的变换，以一种迭加的方式添加到图像块中。李雷达对其进行了一定的改进^[3]，并通过实验验证了水印同步的性能。

SIFT 检测器提取的特征点信息包括坐标、特征尺度、方向和描述符，采用坐标和特征尺度来从图像中确定圆形的局部区域，方法如下：

$$(x - x_0)^2 + (y - y_0)^2 = (k\sigma)^2 \tag{9-22}$$

其中， (x_0, y_0) 表示特征点坐标； σ 为特征尺度； k 为正常数，控制圆的大小。在选取特征点的时候，那些特征尺度很小的特征点需要去掉，因为当图像被缩小的时候这些特征点将很难再检测出来。而且，由于这些特征点的特征尺度太小，生成的圆形区域过小，不利于水印嵌

入。类似地,那些特征尺度过大的特征点也不能被用于水印嵌入,因为最终确定的区域是互不重叠的,如果采用特征尺度过大的特征点,则圆形区域的数量太少,对水印算法的鲁棒性会产生很大的影响。本节中,采用特征尺度范围在 $4\sim 8$ 之间的特征点,这样的话在一幅 512×512 大小的图像上大致可以确定 $6\sim 10$ 个区域(k 取10)。

在水印嵌入端,首先提取 SIFT 特征点,然后筛选出那些特征尺度处于 $4\sim 8$ 之间的特征点。为了在原始图像中确定互不重叠的圆形区域,首先从筛选出来的特征点中选出对应的 DoG 函数值最大的特征点生成圆形区域(因为这个特征点的鲁棒性最好),然后对剩下的特征点进行判断,如果生成的圆形区域与第一个区域重叠,那么就去除这些点。然后再对剩下的特征点进行相同的操作,直到将所有的特征点都处理完,便得到互不重叠的圆形区域。最后,将所有采用的特征点的描述符作为密钥保存下来,用于水印提取。图 9-24 所示为利用上述方法从 Lena 图像中确定的圆形区域($k=10$)^[3]。



图 9-24 基于 SIFT 特征点的水印同步方法确定的圆形区域

水印提取的时候,在受攻击的图像上首先检测 SIFT 特征点,然后将其描述符与水印嵌入时保存下来的描述符利用快速最近邻域法^[32]进行匹配。如果在原始保存的描述符里能找到匹配,那么这个特征点及其特征尺度就用来生成一个圆形区域,且特征尺度的放大倍数与水印嵌入的时候相同。同时需要注意的是,在水印提取的时候考察的特征点特征尺度范围应该更广,因为水印检测时的图像可能被放大或缩小。本节中,考察的特征点特征尺度范围为 $2\sim 16$,这样能保证在图像被缩小一半或放大一倍的时候仍能进行有效的水印提取。

图 9-25 所示为利用上面的方法从不同的图像中确定的圆形区域^[3, 42],可以看到在原始图像中确定的圆形区域和在受攻击图像中确定的圆形区域所包含的图像内容是相同的。



(a) 原始图像

(b) 中值滤波图像

(c) 加噪图像

图 9-25 图像中确定的圆形区域



(d) 30%JPEG 压缩图像 (e) 旋转 30 度的图像 (f) 缩放 0.8 倍的图像

图 9-25 图像中确定的圆形区域（续）

为了进一步对水印同步方案进行性能评价，对 Lena 图像进行常规的信号处理攻击和几何攻击，并分别从中确定水印嵌入和提取所要采用的圆形区域，相匹配的特征区域用相同的颜色进行标记，如图 9-26 所示^[3]。从图中可以看出，当图像受到攻击的时候，在原始图像中生成的绝大部分圆形区域在受攻击后的图像中都能检测出来。正是由于这个原因，如果在这些区域中嵌入水印，则能保证水印能够抵抗这些攻击。同时，也看到有些区域在受攻击图像中不能重新检测出来。例如当图像受到高品质的 JPEG 压缩、噪声、旋转、放大或一些组合攻击的时候。但是由于仍有相当一部分匹配的区域，因此不会对水印检测造成太大的影响。



(a) 原始图像 (b) 中值滤波图像

(c) 20%JPEG 压缩图像 (d) 添加高斯噪声的图像

图 9-26 水印同步性能



(e) 旋转 10 度的图像



(f) 放大 1.2 倍的图像



(g) 平移 40 像素后的图像



(h) 旋转 30 度+缩小 0.8 倍的图像

图 9-26 水印同步性能 (续)

9.5.4 基于 SIFT 特征点的 NSCT 域水印嵌入算法

文献^[3]所提出的图像水印算法的主要思想是在非下采样 Contourlet 变换将水印信号嵌入水印同步过程所提出的局部区域中, 由于从一个图像中可以提取出多个区域, 将相同的水印信号以相同的方式重复地嵌入所有的区域中。水印嵌入的框图如图 9-27 所示^[3]。

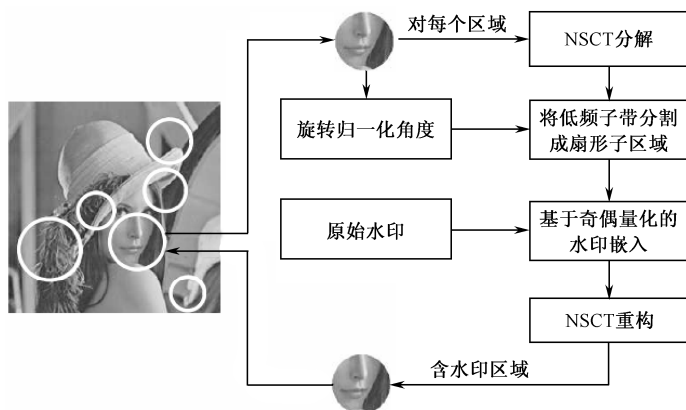


图 9-27 水印嵌入框图

对原始图像，首先利用 9.5.3 节中的方法提取圆形的局部区域，接着进行 NSCT，然后对系数区域进行分割，以满足嵌入多比特水印的要求。然后进行水印嵌入，接着将修改后的系数重构后替代原始的区域，最后得到整幅加水印的图像。在进行水印嵌入之前，还是需要将理想的圆形区域周围补零以得到正方形的图像块，如图 9-28 所示^[3]。类似地，在水印嵌入之后还需要将周围区域的像素去掉，仅采用内接圆内的图像替代原始图像中的相应区域。



图 9-28 理想的圆形区域和正方形区域

下面从三个方面对水印嵌入方法进行介绍，包括嵌入域的选择、圆形区域的分割和基于奇偶量化的水印嵌入。

1. 嵌入域的选择

在基于尺度空间特征点的水印算法中，理想的水印嵌入域也是具有空域和频域局部化特性的变换，如小波变换。但是利用小波变换的一个不足之处是原始的圆形区域经过小波变换后子带图像都较小，不利于水印嵌入。近年来发展起来的多尺度几何分析（Multiscale Geometric Analysis, MGA）理论为这一问题提供了解决的办法。其中，Cunha 等人于 2006 年提出的非下采样 Contouflet 变换（NSCT）由于具有优秀的多分辨率特性、多方向特性和平移不变性，在图像去噪、图像融合等领域得到了广泛的应用。NSCT 在变换时可以根据需要设置所需的方向数，并且分解后的子带和原始图像大小相同。因此比起小波变换，NSCT 对局部区域进行变换产生的频域子带不至于过小而不利于水印信号的嵌入。而且 NSCT 是一种冗余的变换，因此非常适合应用在信息隐藏和数字水印技术中。

2. 圆形区域的分割

对每一个圆形区域，在水印嵌入时首先进行 NSCT，然后在 NSCT 的低频子带中嵌入水印。由于每个圆形区域的大小不同，因此水印嵌入必须依赖于图像的内容。而为了嵌入多比特，需要对圆形区域进行分割。文献[3]提出将 NSCT 低频子带内接圆内的区域进行分割，划分为同心的扇形子区域，分割的示意图如图 9-29 所示^[3]。这些扇形区域具有相同的面积，并且面积的大小由水印序列的长度决定。在得到扇形之后，则将一比特的水印信号通过奇偶量化嵌入一个扇形区域中。

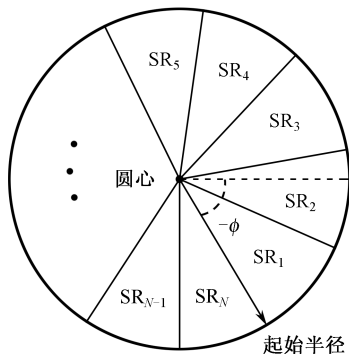


图 9-29 对低频子带内接圆的分割

在对圆形区域进行分割的时候需要考虑图像旋转的影响。当图像发生旋转的时候，圆形区域中的内容虽然相同，但是方向不同。因此为了嵌入水印，必须保证水印嵌入和水印提取的时候分割得到的扇形是一致的。换句话说，水印嵌入时第一个扇形的起始半径位置与水印提取时第一个扇形的起始半径位置相对于图像的内容应该是相同的。这一要求可以通过在水印嵌入和提取前对圆形区域旋转归一化进行，但是那样在水印嵌入后需要进行反向的旋转归一化，因此会产生插值误差，影响水印的检测。为了避免由此产生的插值误差，文献[3]提出了在进行分割的时候仅仅将圆形区域的旋转归一化参数作为参考来确定分割时起始半径的位置，而不进行实际的旋转归一化操作。设圆形区域的旋转归一化角度为 ϕ ，即以 0 度位置的水平半径为参考，顺时针旋转角度作为起始半径位置。

设所要嵌入的二值水印序列为 $w = \{w_1, w_2, w_3, \dots, w_N\}$ ，则每个扇形子区域 (Sub-region, SR) 的角度大小为 $2\pi/N$ 。为了描述每个子区域，圆形区域内的坐标 (x, y) 先变换为极坐标：

$$\begin{cases} \rho_{x,y} = \sqrt{x^2 + y^2} \\ \theta_{x,y} = \arctan(y/x) \end{cases} \quad (9-23)$$

这样，每一个扇形子区域可以表示为

$$SR_i = \{(x, y) | -\phi + (i-1)\frac{2\pi}{N} \leq \theta_{x,y} < -\phi + i\frac{2\pi}{N}\} \quad (9-24)$$

其中， $i=1, 2, \dots, N$ 。需要注意的是极坐标 $\theta_{x,y}$ 的取值范围为 $[0, 360^\circ]$ ，而旋转归一化角度有可能为正也可能为负，因此在利用上式确定扇形区域的时候需要处理一些临界情况。设每一个扇形子区域 SR_i 的起始角度为 φ_i^1 ，结束角度为 φ_i^2 。由上式知 $\varphi_i^1 = -\phi + (i-1)\frac{2\pi}{N}$ ， $\varphi_i^2 = -\phi + i\frac{2\pi}{N}$ ，临界条件的处理也就是保证 $\varphi_i^1 \in [0, 360^\circ]$ ， $\varphi_i^2 \in [0, 360^\circ]$ 。具体的调整过程和扇形区域的确定方法如下。

%BlkImg 表示提取的圆形图像块， SR_i 表示分割出来的扇形子区域。

情况 1：旋转归一化角度 $\phi > 0$ 的情况^[3]。

```
for i=1: N
```

```
if  $\varphi_i^1 < 0$  &  $\varphi_i^2 < 0$ 
```

```
     $\varphi_i^1 = \varphi_i^1 + 360$ 
```

```
     $\varphi_i^2 = \varphi_i^2 + 360$ 
```

% N 表示扇形子区域的个数，即水印序列的长度

% $\varphi_i^1 \notin [0, 360^\circ]$, $\varphi_i^2 \notin [0, 360^\circ]$ 的情况

% 将 φ_i^1 调整至 $[0, 360^\circ]$ 区间

% 将 φ_i^2 调整至 $[0, 360^\circ]$ 区间

```
SRi=BlkImg.* (  $\theta_{x,y} \geq \varphi_i^1$  &  $\theta_{x,y} \leq \varphi_i^2$  ) ;    % .*表示点乘; &表示按位与
elseif  $\varphi_i^1 < 0$  &&  $\varphi_i^2 > 0$                         %  $\varphi_i^1 \notin [0, 360^\circ]$ ,  $\varphi_i^2 \in [0, 360^\circ]$  的情况
     $\varphi_i^1 = \varphi_i^1 + 360$                                 % 将  $\varphi_i^1$  调整至  $[0, 360^\circ]$  区间
SRi=BlkImg.* (  $\theta_{x,y} \geq \varphi_i^1$  |  $\theta_{x,y} \leq \varphi_i^2$  ) ;    % | 表示按位或
else                                                    % 一般情况
    SRi=BlkImg.* (  $\theta_{x,y} \geq \varphi_i^1$  &  $\theta_{x,y} \leq \varphi_i^2$  ) ;
end
end
```

情况 2: 旋转归一化角度 $\phi < 0$ 的情况^[3]。

```
for i=1: N
if  $\varphi_i^1 > 360$  &  $\varphi_i^2 > 360$                         %  $\varphi_i^1 \notin [0, 360^\circ]$ ,  $\varphi_i^2 \notin [0, 360^\circ]$  的情况
     $\varphi_i^1 = \varphi_i^1 - 360$                                 % 将  $\varphi_i^1$  调整至  $[0, 360^\circ]$  区间
     $\varphi_i^2 = \varphi_i^2 - 360$                                 % 将  $\varphi_i^2$  调整至  $[0, 360^\circ]$  区间
SRi=BlkImg.* (  $\theta_{x,y} \geq \varphi_i^1$  &  $\theta_{x,y} \leq \varphi_i^2$  ) ;
elseif  $\varphi_i^1 < 360$  &&  $\varphi_i^2 > 360$                     %  $\varphi_i^1 \in [0, 360^\circ]$ ,  $\varphi_i^2 \notin [0, 360^\circ]$  的情况
     $\varphi_i^2 = \varphi_i^2 - 360$                                 % 将  $\varphi_i^2$  调整至  $[0, 360^\circ]$  区间
SRi=BlkImg.* (  $\theta_{x,y} \geq \varphi_i^1$  |  $\theta_{x,y} \leq \varphi_i^2$  ) ;
else                                                    % 一般情况
    SRi=BlkImg.* (  $\theta_{x,y} \geq \varphi_i^1$  &  $\theta_{x,y} \leq \varphi_i^2$  ) ;
end
end
```

图 9-30 给出了扇形子区域分割的一个例子^[3]。设所要嵌入的水印序列长度为 6，图 9-30 最左边一列为分别从原始图像、旋转 30 度图像和旋转 90 度图像中提取的同样一块区域，图 9-30 的第二列至第七列为利用文献[3]所提出的算法分割出的等面积扇形区域。由图可知，从原始图像和旋转后的图像中提取出的区域具有缩放不变性。即所包含的图像内容是相同的，但是三个图像块在方位上存在差异。由第二列至第七列可知，虽然原始的图像块方向各不相同，但是分割所得到的扇形区域内容是一样的。如第五列，都是 Lena 图像中嘴巴区域的内容，并且内容完全相同。因此，通过这样的方式将水印依次嵌入每一个扇形区域中，则在受到旋转、缩放等几何攻击的时候就可保证每一位水印嵌入和提取是在相同的位置进行的，这样就保证了水印检测的鲁棒性。

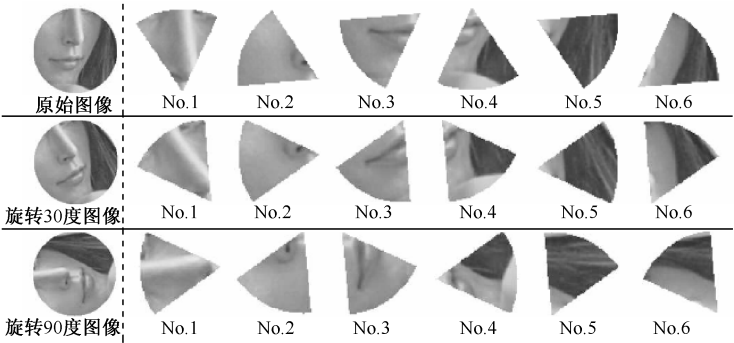


图 9-30 具有旋转不变性的扇形区域分割

3. 基于奇偶量化的水印嵌入

文献[3]通过奇偶量化将水印信号在 NSCT 域嵌入每一个扇形子区域中, 并且在嵌入的时候水印比特的序号和扇形子区域的序号相对应。对水印 $w = \{w_1, w_2, w_3, \dots, w_N\}$, 也就是将 w_i 嵌入子区域 SR_i 中。具体嵌入的时候, 根据所要嵌入的水印比特, 将每个子区域中的 NSCT 系数进行奇偶量化, 使其具有相同的奇偶特性。对子区域中的每一个 NSCT 系数 $c(x, y)$, 首先利用量化函数赋符号“0”或“1”:

$$Q(x, y) = \begin{cases} 0, & k\Delta \leq c(x, y) < (k+1)\Delta, \quad k = 0, \pm 2, \pm 4, \dots \\ 1, & k\Delta \leq c(x, y) < (k+1)\Delta, \quad k = \pm 1, \pm 3, \pm 5, \dots \end{cases} \quad (9-25)$$

其中, Δ 表示量化步长。为了提高水印嵌入的鲁棒性, 修改后的 NSCT 系数应该处于相应量化间隔的中间。为此, 首先计算每个系数 $c(x, y)$ 的量化噪声:

$$r(x, y) = c(x, y) - \left\lfloor \frac{c(x, y)}{\Delta} \right\rfloor \cdot \Delta \quad (9-26)$$

其中, $\lfloor \cdot \rfloor$ 为下取整操作。系数 $c(x, y)$ 的修改量 $u(x, y)$ 由下式确定:

$$u(x, y) = \begin{cases} -r(x, y) + 0.5\Delta, & Q(x, y) = w_i \\ -r(x, y) + 1.5\Delta, & Q(x, y) \neq w_i, \quad r(x, y) > 0.5\Delta \\ -r(x, y) - 0.5\Delta, & Q(x, y) \neq w_i, \quad r(x, y) \leq 0.5\Delta \end{cases} \quad (9-27)$$

最后, 修改后的系数值 $c^*(x, y)$ 为

$$c^*(x, y) = c(x, y) + u(x, y) \quad (9-28)$$

上述修改操作对每一个扇形子区域内的所有 NSCT 系数进行, 便得到一个圆形区域修改后的 NSCT 系数。对修改后的 NSCT 系数进行重构, 便得到含水印的圆形区域, 然后用其替换原始图像中的对应区域。对所有的圆形区域进行相同的水印嵌入操作, 最后便得到整幅含水印的图像。

9.5.5 基于 SIFT 特征点的 NSCT 域水印提取算法

水印提取算法的过程如图 9-31 所示^[3]。

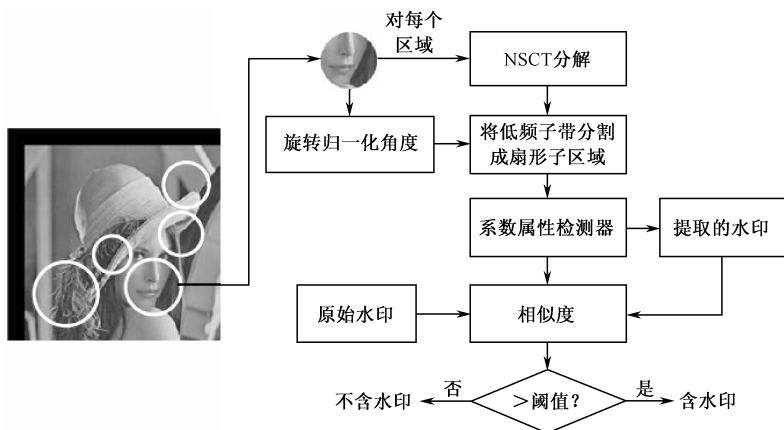


图 9-31 水印提取算法的过程

对受攻击图像,首先利用 9.5.3 节中的方法在受攻击图像中定位嵌入了水印的区域,然后利用与嵌入过程相同的方法对每个区域进行分割,接着进行水印提取并通过比较与原始水印信号的相似度来判断水印存在与否。

1. 系数属性检测器

由于水印嵌入是在 NSCT 域进行的,所以水印提取也在 NSCT 域进行。文献[3]提出了一种系数属性检测器 (Coefficient Property Detector, CPD) 来提取水印。具体操作时,把一个圆形区域首先变换到 NSCT 域,并依据旋转归一化角度分割成等面积的扇形区域。然后对每个扇形子区域内的系数进行量化,并记量化值为“0”的系数个数为 $NUM_{i,0}$, 量化值为“1”的系数个数为 $NUM_{i,1}$, 则水印提取方程为

$$w'_i = \begin{cases} 0, & NUM_{i,0} > NUM_{i,1} \\ 1, & NUM_{i,1} > NUM_{i,0} \end{cases} \quad (9-29)$$

其中, $w' = \{w'_1, w'_2, w'_3, \dots, w'_N\}$ 表示提取的水印信号。最后, 计算原始水印信号 w 和提取的水印信号 w' 的归一化汉明相似度 (Normalized Hamming Similarity, NHS)。

$$NHS = 1 - \frac{HD(w, w')}{N} \quad (9-30)$$

其中, $HD(\cdot)$ 表示两个序列的汉明距离, N 表示水印序列的长度。易知, NHS 的最大值为 1, 并且 NHS 值越大, 提取的水印信号越接近于原始的水印。

2. 虚警概率分析

要判断水印存在与否, 须将 NHS 值与一阈值比较。借助于文献[43]计算虚警概率的方法来确定这一阈值。设水印序列的每比特为独立的变量, 则在 N 比特长的提取水印中, 有 k 比特与原始水印正确匹配的概率为

$$P_k = \binom{N}{k} p^k (1-p)^{N-k} \quad (9-31)$$

其中, p 表示提取水印的每一位与原始水印的对应位正确匹配的概率。由于对应的两个比特正确匹配的概率 $p=0.5$, 故 P_k 可以写为:

$$P_k = (0.5)^N \frac{N!}{k!(N-k)!} \quad (9-32)$$

于是, 每一个圆形区域中提取水印的虚警概率与设定的阈值之间的关系为

$$P_{\text{local}} = \sum_{k=T}^N (0.5)^N \frac{N!}{k!(N-k)!} \quad (9-33)$$

其中, k 为正确匹配的位数, T 为检测的阈值。设定若能从两个区域中成功提取水印则表明水印存在。故从整幅图像中提取水印的虚警概率为

$$P_{\text{global}} = \sum_{i=2}^M (P_{\text{local}})^i (1-P_{\text{local}})^{M-1} \binom{M}{i} \quad (9-34)$$

其中, M 表示圆形区域的个数。

图 9-32 所示为水印序列长度为 32 比特的情况下, 设置不同的检测阈值时的虚警概率曲

线 ($M=6$)^[3]。从图中可以看出,随着检测阈值的增大,虚警概率逐渐减小。可以根据不同的虚警概率要求设置不同的水印检测阈值。

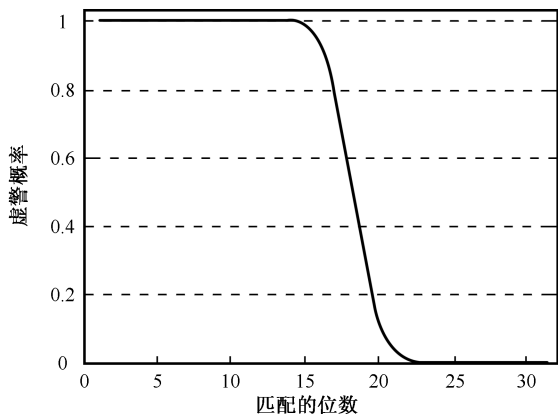


图 9-32 不同阈值的虚警概率曲线

9.5.6 实验结果与讨论

文献[3]实验中所采用的图像为 512×512 的灰度图像,包括 Lena、Peppers、Baboon 和 GoldHill 等。实验中,水印序列的长度为 32 比特,水印检测的阈值设置为 24,对应的 NHS 阈值为 0.75,这时候水印系统的虚警概率为 1.8×10^{-4} 。

实验中,还与 Tang 等人的算法^[44]、Wang 等人的算法^[45]的性能进行了比较。

Tang 等人利用墨西哥帽小波的尺度交互特性进行图像特征点的提取,并选取其中的一些特征点作为参考生成圆形的区域,然后结合图像归一化技术在离散傅里叶变换(DFT)域嵌入二值水印序列^[44]。

Wang 等基于 Harris-Laplace 特征点提出了一种 DFT 域的水印嵌入和提取算法,其基本思想是用 Harris-Laplace 特征点进行水印同步。

1. 水印不可见性

由于采用奇偶量化嵌入水印,需要确定合适的量化步长,以便达到水印不可见性和鲁棒性的折中。

图 9-33 所示为在 Lena (平滑图像) 和 Baboon (纹理复杂) 图像上采取不同的量化步长时图像的 PSNR 值曲线^[3]。可以发现,随着量化步长的增大,图像的 PSNR 值越来越小。当 Δ 小于 0.025 的时候,两个图像的 PSNR 值都在 40dB 以上;当 Δ 接近 0.06 的时候,PSNR 值接近 30dB。当 Δ 处于 0.03~0.04 之间时,PSNR 值仍高于 35dB,但此时实验中发现 Lena 和 Peppers 等平滑图像嵌入水印的区域会产生较大的失真。综合考虑以上因素,设置实验中所采用的奇偶量化步长为 0.02。此时加水印图像的 PSNR 值都在 40dB 以上且嵌入水印的局部区域也不会有明显的失真。从图 9-33 中还可看出,所嵌入的水印长度对图像 PSNR 值的影响很小。

图 9-34 为水印嵌入的效果^[3]。图 9-34 (a) 和图 9-34 (b) 分别表示原始图像和加水印图像;图 9-34 (c) 为放大 100 倍后的差值图像。从图中可以看出,水印嵌入对图像质量的影响

很小，人眼觉察不出明显的差别。四幅图像水印嵌入的圆形区域数目分别为 6、6、7 和 7，PSNR 值分别为 44.039dB、45.282dB、45.578dB 和 44.974dB。文献[3]的算法所获得的图像质量与 Tang 等人^[44]的算法相当，且两者都比 Lee 等人的算法^[42]好。这主要是因为 Lee 等人的算法通过修改空域像素嵌入水印，而 Tang 等人的算法与文献[3]的算法都在变换域嵌入水印。

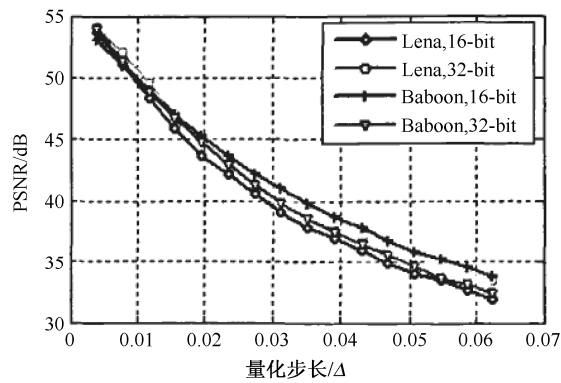
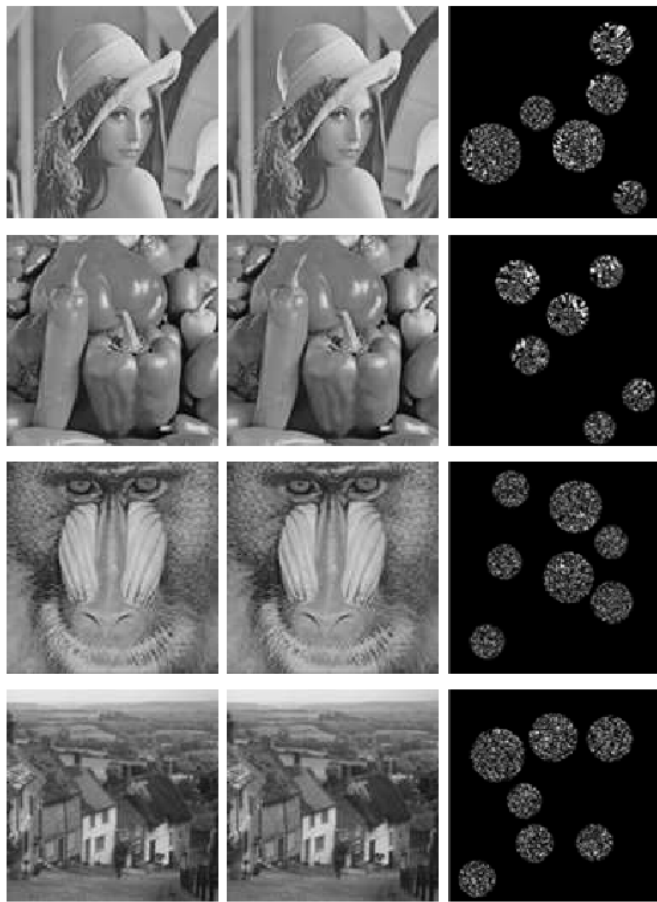


图 9-33 量化步长与图像的PSNR值之间的关系



(a) 原始图像 (b) 加水印图像 (c) 差值图像

图 9-34 水印嵌入效果

2. 水印鲁棒性

从水印抵抗常规的信号处理攻击和几何攻击两个方面的能力考察水印的鲁棒性。常规的信号处理攻击包括中值滤波、添加高斯噪声、锐化和 JPEG 压缩，几何攻击包括 RST 攻击、行/列删除以及一些组合的攻击。实验结果如表 9-1 所示，其中分母表示水印嵌入过程中用到的圆形区域数目，分子表示水印提取过程中能正确提取水印的区域数目。为了进行比较，文献[3]同时给出了 Tang 的算法^[44]和 Wang 的算法^[45]的实验结果。

表 9-1 正确提取水印的区域数与原始区域数的比率

攻击类型	Lena			Peppers			Baboon		
	本书	Wang	Tang	本书	Wang	Tang	本书	Wang	Tang
中值滤波（3×3）	4/6	3/6	1/8	5/6	4/8	1/4	2/7	7/12	2/11
锐化（3×3）	6/6	3/6	3/8	6/6	5/8	2/4	4/7	6/12	4/11
添加高斯噪声	3/6	2/6	2/8	4/6	4/8	2/4	5/7	4/12	3/11
JPEG 压缩 30%	4/6	2/6	2/8	4/6	4/8	0/4	2/7	8/12	4/11
JPEG 压缩 50%	4/6	4/6	4/8	6/6	6/8	2/4	6/7	8/12	6/11
JPEG 压缩 70%	4/6	4/6	5/8	6/6	7/8	3/4	6/7	9/12	8/11
中值滤波+ JPEG 90%	5/6	3/6	1/8	4/6	4/8	1/4	3/7	7/12	1/11
锐化+ JPEG 90%	6/6	3/6	3/8	6/6	6/8	3/4	4/7	6/12	2/11
删除 8 行 16 列	4/6	5/6	1/8	4/6	6/8	0/4	5/7	7/12	2/11
剪切 55%	2/6	4/6	1/8	2/6	6/8	0/4	1/7	6/12	2/11
旋转 5 度	5/6	4/6	3/8	6/6	5/8	1/4	4/7	5/12	3/11
旋转 15 度	4/6	3/6	1/8	6/6	5/8	0/4	4/7	4/12	2/11
旋转 30 度	4/6	2/6	0/8	5/6	4/8	0/4	2/7	4/12	0/11
平移 10 像素	4/6	5/6	2/8	4/6	2/8	1/4	6/7	10/12	8/11
缩放 0.6 倍	1/6	1/6	0/8	4/6	5/8	1/4	1/7	2/12	1/11
缩放 0.9 倍	4/6	3/6	1/8	4/6	3/8	1/4	2/7	5/12	2/11
缩放 1.4 倍	3/6	1/6	0/8	3/6	3/8	0/4	2/7	1/12	0/11
剪切 10%+ JPEG 70%	4/6	2/6	2/8	3/6	2/8	1/4	5/7	5/12	2/11
旋转 5 度+缩放 0.9 倍	4/6	3/6	0/8	4/6	4/8	0/4	2/7	5/12	1/11
平移 10 像素+旋转 5 度 +缩放 0.9 倍	3/6	2/6	0/8	3/6	3/8	0/4	2/7	3/12	1/11

为了更直观地对算法进行性能评价，表 9-2 给出了水印提取的相似度，包括相似度的最大值和平均值。

表 9-2 提取水印与原始水印的相似度

攻击类型	Lena		Peppers		Baboon	
	HNS _{max}	NHS _{mean}	HNS _{max}	NHS _{mean}	HNS _{max}	NHS _{mean}
添加高斯噪声	0.969	0.906	0.969	0.891	0.875	0.819
中值滤波（3×3）	1.000	0.969	1.000	0.906	0.938	0.860
中值滤波（5×5）	0.938	0.852	0.969	0.917	0.781	0.781
JPEG 压缩 30%	0.939	0.900	0.938	0.867	0.813	0.813
JPEG 压缩 50%	1.000	0.969	1.000	0.932	0.969	0.860

续表

攻击类型	Lena		Peppers		Baboon	
	HNS _{max}	NHS _{mean}	HNS _{max}	NHS _{mean}	HNS _{max}	NHS _{mean}
JPEG 压缩 70%	1.000	0.969	1.000	0.958	1.000	0.980
中值滤波 (3×3) +JPEG90	1.000	0.988	1.000	0.906	0.969	0.833
旋转 10 度	0.969	0.963	1.000	0.975	1.000	0.866
旋转 30 度	0.969	0.922	0.969	0.919	1.000	0.906
旋转 45 度	0.906	0.891	0.969	0.888	1.000	0.922
旋转 60 度	0.906	0.850	0.969	0.896	0.938	0.875
旋转 90 度	0.969	0.844	0.969	0.869	0.813	0.813
旋转 10 度+剪切	1.000	0.969	1.000	0.956	1.000	0.917
旋转 30 度+剪切	0.969	0.922	1.000	0.980	1.000	0.927
旋转 60 度+剪切	0.938	0.917	0.969	0.885	0.906	0.828
缩放 0.6 倍	0.844	0.844	0.938	0.898	0.750	0.750
缩放 0.8 倍	1.000	0.888	1.000	0.969	0.938	0.906
缩放 1.2 倍	1.000	0.960	1.000	0.980	1.000	0.860
缩放 1.4 倍	1.000	0.969	1.000	0.906	0.875	0.875
平移 40 像素	1.000	1.000	1.000	1.000	1.000	0.969
旋转 5 度+缩放 0.9 倍	0.969	0.938	0.969	0.906	0.906	0.828
旋转 10 度+缩放 0.9 倍	1.000	0.938	1.000	0.938	0.969	0.896
旋转 15 度+缩放 0.9 倍	1.000	0.948	1.000	0.938	0.906	0.906
旋转 30 度+缩放 0.9 倍	0.906	0.875	0.969	0.919	0.969	0.875

(1) 信号处理攻击

从表 9-1 和表 9-2 中可以看出，文献[3]提出的算法对常规的各种图像处理操作都具有很好的检测性能。尤其是对 JPEG 压缩攻击，即使当品质因数降至 30%的时候也可以很可靠地进行水印提取。该算法对信号处理攻击的鲁棒性与 Wang 等人的算法相当，且都优于 Tang 等人的算法。

(2) 旋转攻击

对于图像的旋转，文献[3]的算法可以直接从旋转后的图像中进行水印检测，而不需要其他辅助操作。这主要是因为该算法在水印嵌入和水印提取的时候对圆形区域的分割本身具有旋转不变性。在 Wang 等人的算法中，当图像发生旋转的时候需要首先获得旋转的角度，然后对旋转进行校正。而校正的过程必然携带着图像的插值，会对水印检测造成进一步的影响。Tang 等人的算法仅能抵抗小的旋转，通常小于 5 度；Wang 等人的算法可以抵抗稍大的旋转，如 30 度。相比较而言，文献[3]的算法对旋转的鲁棒性最好，可以抵抗如 60 度或 90 度这样的大角度旋转，而且提取水印的相似度都在 0.8 以上。

(3) 缩放攻击

对基于特征的水印算法，水印的缩放不变性主要由两方面的因素决定：

- ① 水印的嵌入区域应当具有缩放不变性，即图像缩放的时候局部区域内的图像内容是相同的；
- ② 水印的嵌入以一种内容自适应（Content Based Manner）的方式进行。在文献[3]的算

法中,基于 SIFT 的水印同步方案中确定的圆形区域满足第一个条件,而将水印嵌入扇形区域中满足了第二个条件。Wang 等的算法^[45]和文献[3]的算法都可以抵抗缩放攻击,而 Tang 等的算法^[44]不能。这是因为 Tang 的算法中基于墨西哥帽小波的特征点提取方法不具有缩放不变性。

(4) 平移攻击

在图像发生平移的时候,一部分的图像内容会被丢失。但从剩下的图像内容中仍可以检测出嵌入了水印的区域,所以仍然可以准确地提取水印,而且剩余区域的水印相似度都非常高,因为剩下的区域并没有受到任何攻击。

(5) 其他的几何攻击

除了 RST 攻击,文献[3]还对其他类型的几何攻击进行了测试,如图像剪切和行/列的删除,实验结果表明水印检测性能都很好。对于剪切攻击,Wang 等的算法^[45]性能稍优于文献[3]的算法,因为提取的区域数目更多,所以剪切后会剩下更多的区域。如果文献[3]的算法采用较小的圆形区域半径,则算法对剪切的鲁棒性会有很大的提高。

(6) 组合攻击

组合攻击主要包括旋转加剪切、旋转加缩放、中值滤波加 JPEG 压缩等。从表 9-1 和表 9-2 中可以看出,在受到这些攻击后水印都能够成功地提取。

综上所述,本节所介绍的基于 SIFT 特征点的 NSCT 域图像水印算法^[3]可以很好地抵抗常规的信号处理攻击、几何攻击以及一些组合攻击,在水印的不可见性和鲁棒性方面都取得了很好的效果。但该算法的一个缺点就是水印的容量较小,一般为几十比特。文献[3]在 SIFT 特征点水印同步的基础之上,提出了一种基于 DWT 域的大容量鲁棒图像水印算法。

基于图像特征点的水印算法优点是可以获得很高的图像质量,因为这类算法水印是嵌入图像的局部区域中的,而不是整个图像中。另外,水印鲁棒性较好,可以抵抗图像的 RST 攻击以及剪切攻击。缺点是在水印检测之前需要进行同步操作(即重新检测局部区域),因此会增加整个算法的运算量。并且,算法性能的优劣很大程度上取决于特征点检测的稳定性。

除了上述几种类型的抗几何攻击数字水印算法之外,还有一些其他类型的算法,如基于水印自同步(Self-synchronzation)特性的方法、基于直方图的方法、基于神经网络与支持向量机的算法等。此外除了抵抗全局几何攻击外,还出现了针对局部几何失真的算法。

参考文献

- [1] M. Kutter, S. K. Bhattacharjee, T. Ebrahimi. Towards Second Generation Watermarking Schemes, in Proceedings of International Conference on Image Processing, 1999, 1: 320-323.
- [2] D. Zheng, Y. Liu, J. Zhao, et al. A Survey of RST Invariant Image Watermarking Algorithms, ACM Computing Surveys, 2007, 39(2): 1-91.
- [3] 李雷达. 水印抗几何攻击理论及应用研究. 西安电子科技大学博士学位论文, 2009.
- [4] D. Delannay. Digital Watermarking Algorithms Robust Against Loss of Synchronization, Ph. D Thesis, Université catholique de Louvain, 2004.

- [5] V. Licks, R. Jordan. Geometric attacks on image watermarking systems, IEEE Multimedia, 2005, 12(3): 68-78.
- [6] W. Agung, P. Sweeney. Method for combating random geometric attack on image watermarking. Electronics Letters. 2008(7):420-421.
- [7] J. Du, C. S. Woo, B. Pham. Recovery of watermark using differential affine motion estimation. Australasian Information Security Workshop. 2005.
- [8] J. F. Lichtenauer, I. Setyawan, T. Kalker, et al. False positive watermark detection behavior in exhaustive geometrical searches. Ninth annual conference of the Advanced School for Computing and Imageing. 2003.
- [9] D. Simitopoulos, D. E. Koutsonanos, M. G. Strintzis Robust image watermarking based on generalized randon transformation. IEEE Transaction on circuits and systems for vidio technology. 2008(8):732-745.
- [10] D. Simitopoulos, D. Koutsonanos, M. G. Strintzis. Image watermarking resistant to geometric attacks using generalized radon transformations. IEEE Transaction on DSP. 2002:85-88.
- [11] L. G. Brown. A survey of image registration techniques. ACM Computing Surveys, 1992, 24(4): 325-376.
- [12] G. W. Braudaway, F. Mintzer. Automatic recovery of invisible image watermarks from geometrically distorted images. Journal of Electronic Imaging, 2000, 9(4):477-483.
- [13] P. Dong, J.G. Brankov, N. Galatsanos, et al. Geometric robust watermarking based on a new mesh model correction approach. in: International Conference on Image Processing (ICIP'02), Rochester, NY, United States, 2002, 3: 493-496.
- [14] I. B. Ozer, M. Ramkumar, A. N. Akansu. A new method for detection of watermarks in geometrically distorted images. in: 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'00), Istanbul, Turkey, 2000, 4: 1963-1966.
- [15] H. Cheng. A review of video registration methods for watermark detection in digital cinema applications. in: 2004 IEEE International Symposium on Circuits and Systems, Vancouver, BC, Canada, 2004, 5: 704-707.
- [16] 余艳玮. 基于特征点的抗几何攻击的图像盲水印技术研究. 华中科技大学博士学位论文, 2007.
- [17] M. Barni. Effectiveness of Exhaustive Search and Template Matching Against Watermark Desynchronization, IEEE Signal Processing Letters, 2005, 12(2): 158-161.
- [18] J. J. K. O'Ruanidh, T. Pun. Rotation, Scale and Translation Invariant Digital Image Watermarking. in Proceedings of IEEE International Conference on Image Processing, 1997: 636-539.
- [19] D. Zheng, J. Y Zhao. LPM-based RST Invariant Digital Image Watermarking, in Proceedings of Canadian Conference on Electrical and Computer Engineering, 2003,3:1951-1954.
- [20] C.Y. Lin, M. Wu, J.A. Bloom, et al. Rotation, Scale, and Translation Resilient Watermarking for Images, IEEE Transactions on Image Processing, 2001, 10(5): 767-782.

- [21] D. Zheng, J.Y. Zhao, A. El Saddik. RST-invariant Digital Image Watermarking Based on Log-Polar Mapping and Phase Correlation, IEEE Transactions on Image Processing, 2003, 13(8): 753-765.
- [22] H. S. Kim, H. Y. Lee. Invariant Image Watermark Using Zernike Moments, IEEE Transactions on Circuit and Systems for Video Technology, 2003, 13(8):766-775.
- [23] Y. Q. Xin, S. Liao, M. Pawlak. A Multibit Geometrically Robust Image Watermark Based on Zernike Moments. in Proceedings of International Conference On Pattern Recognition, 2004, 4: 861-864.
- [24] Y. Q. Xin, S. Liao, M. Pawlak. Geometrically Robust Image Watermarking via Pseudo-Zernike Moments. in Proceedings of Canadian Conference on Electrical and Computer Engineering, 2004, 2: 939-942.
- [25] Y. Q. Xin, S. Liao, M. Pawlak. Robust Data Hiding With Image Invariants. in Proceedings of Canadian Conference on Electrical and Computer Engineering, 2005: 963-966.
- [26] Y. Q. Xin, S. Liao, M. Pawlak. Circularly Orthogonal Moments for Geometrically Robust Image Watermarking. Pattern Recognition, 2007, 40(12): 3740-3752.
- [27] M. Alghoniemy, A. H. Tewfik. Geometric Invariance in Image Watermarking, IEEE Transactions on Image Processing, 2004, 13(2): 145-153.
- [28] S. M. Smith, J. M. Brady. SUSAN-A New Approach to Low Level Image Processing, International Journal of Computer Vision, 1997, 23(1): 45-78.
- [29] C. Harris, M. Stephens. A Combined Corner and Edge Detector, in Proceedings of the 4th Alvey Vision Conference, 1988, 147-151.
- [30] K. Mikolajczyk, C. Schmid. Indexing Based on Scale Invariant Interest Points, in Proceeding of the 8th International Conference on Computer Vision, 2001, 525-531.
- [31] K. Mikolajczyk, C. Schmid. Scale and Affine Invariant Interest Point Detectors, International Journal of Computer Vision, 2004, 60(1): 63-86.
- [32] D.G. Lowe. Distinctive Image Features from Scale-invariant Keypoints, International Journal of Computer Vision, 2004, 60(2): 91-110.
- [33] T. Lindeberg. Scale-Space Theory: A Basic Tool for Analysing Structures at Different Scales, Journal of Applied Statistics, 1994, 21(2): 224-270.
- [34] P. Bas, J. M. Chassery, B. Macq. Geometrically Invariant Watermarking Using Feature Points, IEEE Transactions on Image Processing, 2002, 11(9): 1014-1028.
- [35] H. P. Moravec. Obstacle avoidance and navigation in the real world by a seen robot rover. Carnegie-Mellon University, Technology Report CMU-RI-TR-3, 1980.
- [36] 周知. 三角剖分算法研究. 哈尔滨工业大学硕士学位论文, 2007.
- [37] 李健. 抗几何攻击的数字图像水印技术的研究. 南京理工大学博士学位论文, 2009.
- [38] 孟亮, 方金云, 唐志敏. Delaunay 三角网表示和点删除方法. 计算机工程与设计, 2008 (3): 738-741.
- [39] 李水乡, 陈斌, 赵亮, 等. 快速 Delaunay 逐点插入网格生成算法. 北京大学学报 (自然科学版), 2006 (3): 1-5.

- [40] S. W. Sloan. A fast algorithm for generating constrained Delaunay triangulation. In Computers and Structures. 1993: Pergammon Press Ltd.441-450.
- [41] D. G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision. 2004 (2) :91-110.
- [42] H. Y. Lee, H. S. Kim, H. K. Lee. Robust Image Watermarking Using Local Invariant Features, Optical Engineering, 2006, 45 (3): 1-11.
- [43] X. J. Qi, J. Qi. A Robust Content-based Digital Image Watermarking Scheme, Signal Processing, 2007, 87 (6): 1264-1280.
- [44] C. W. Tang, H. M. Hang. A Feature-based Robust Digital Image Watermarking Scheme, IEEE Transactions on Signal Processing, 2003, 51 (4): 950-959.
- [45] X. Wang, J. Wu, P. Niu. A New Digital Image Watermarking Algorithm Resilient to Desynchronization Attacks, IEEE Transactions on Information Forensics and Security, 2007, 2 (4): 655-663.